SIMULATION OF OPTIMIZATION IN NETWORKS

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

Master of Philosophy

IN

OPERATIONS RESEARCH

BY

MOHAMMAD RIZWANULLAH

Under the Supervision of

DR. QAZI MAZHAR ALI

DEPARTMENT OF STATISTICS AND OPERATIONS RESEARCH
ALIGARH MUSLIM UNIVERSITY
ALIGARH (INDIA)

2003

Dedicated to

My

Beloved Parents

Dr. Qazi Mazhar Ali
D. Phil. (Oxon.)

Department of
Statistics & Operations Research
Aligarh Muslim University
Aligarh, UP, 202002, India
Phone (0571) - 2701251(O)
(0571) - 2720158(R)

Dated: - ... ...

Certificate

This is to Certify that the material contained in the dissertation entitled ***"Simulation of Optimisation in Networks"*** being submitted by **Mr. Mohammad Rizwanullah,** Deptt. of Statistics and Operations Research, A.M.U., Aligarh for the award of the Master of Philosophy in Operations Research is a record of his own work carried out by him under my supervision and guidance.

In my opinion, it is worthy of consideration for the award of a M. Phil degree in "**Operations Research**".

20/5/03

(Dr. Qazi Mazhar Ali)
Supervisor

Contents

ACKNOWLEDGMENT

All praises and thanks are for almighty '**Allah**', who have enabled me to complete this research work in the present form.

It gives me extreme pleasure to place on record my indebtedness and sincere gratitude to my research supervisor, ***Dr. Qazi Mazhar Ali*** (Reader), Department of Statistics and Operations Research, Aligarh Muslim University, for helpful comments, suggestion, encouragement, invaluable guidance and research facilities. I confess that without his good advice and help this would not have been possible.

I have the pleasant duty of offering my thanks to **Prof. Serajur Rahman** (Chairman), Prof. Sanaullah Khan for the inspiration and valued help that received from them during the course of this study. I am highly thankful to Prof. M.Z.Khan, Dr. M.J. Ahsan, Dr. Abdul Bari, Dr. Kamaullah and other teaching and non-teaching staff members of the Department.

Further, I like to express my sincere regards to Mr. Alsad, Lecturer, Deptt. of Computer Engg. Mr. Afzal Ahmad, System Engineer, Computer Centre, Prof. Azeemuddin Qureshi, Chairman Deptt. of Geography, A.M.U., and his staff for their generous help in collecting and processing the data.

I thank to all my friends, specifically, Mr. Mohd. Asif, Mr. Teg Alam, Mr. Shahid Ali Khan, Mr. Jan Mir, Mr. Waseem, Ms. Salma, Mr. Hamid, Ms. Rifat Tabassum and Ms. Saba Khan.

This acknowledgment will be incomplete if I fail to express my deep sense of obligation and severance to my parents and brothers *Mr. Mohammad. Ahsanullah & Mr.Mohammad Saifullah Khalid* whose moral and financial support enabled me to complete this work.

Md Rizwanullah

(*Mohammad. Rizwanullah*)

Department of Statistics and Operations Research
Aligarh Muslim University
Aligarh. U.P., India.

Chapter-1

A 'network' is a comprehensive programme of activities which permits the manager of the system to visualize the many facets of the scheme as an integrated whole. It is composed of all the relevant events and activities necessary for the accomplishment of the objectives of a project and displays graphically these events and activities, their sequences, concurrencies and inter – relationships.

There are many problems in real life situations such as:

- The Construction of a building,
- Planning and launching a new project,
- The manufacturing and assembling of a large machine tool,
- Preparation of budget for a company,
- Communication system,
- Fluid supply system,
- Airways system etc,

which can be viewed as network problems. There are many physical systems or real life systems for which it becomes rather complex to build a mathematical model. Even if the mathematical model can be constructed to a reasonable degree of accuracy, there are not available the procedures to solve the resulting model. The other means are resorted to analyze the system. One such means is simulation. The variety of models which are studied through simulation are scale models of airplanes tested in wind tunnels, a hydraulic or a mechanical circuit converted into an analogous electric circuit, inventory, queuing, scheduling and forecasting.

Several networks arising in the vast campus of Aligarh Muslim University, have been considered in this dissertation. The methods of solution for the respective network problems have also been discussed.

The Software packages used for solution of the actual problems are (i) LINGO and (ii) TORA.

Flow Net theory was developed considerably since the beginning of 1950's. Networks are diagrams, easily visualized in Electrical theory, Transportation systems like Roads, Railway lines, Pipelines, Blood vessels. In Operations Research, Network play an important role as quite often the problem of determining an optimum solution can be looked upon as the problem of selecting the best sequence of operation out of a finite number of available alternation that can be represented as a Network.

The "Network" provides a patent technique for programming and scheduling activities and resources (financial, material and human); determining time, resources and cost status; Schedule step pages; Resource overloads and Cost over runs; and developing possibilities of effective corrective action.

It thus provides an integrated framework for effective management decision at all stages. The principal steps are as:

(a) **Programming**: i.e. setting up the network of phased activities giving the estimated activity and project duration, estimated material and manpower requirements.

(b) **Scheduling**: i.e. estimating integrated and feasible time scheduled of activities, financing and resource development.

(c) **Progressing**: i.e. determining project status of works, evaluation and forecasting future status on the basis of present status and subsequent corrective action.

(d) **Modifying**: i.e. making necessary adjustments and revision in the network, activity-wise, time-wise, resource-wise, etc. to secure expedition, economy and efficiency.

The family of network optimization problems includes the following prototype models: assignment, critical path, max flow, shortest path, transportation and minimum cost flow problems. These arcs easily stated by using a network of arcs and nodes.

A network consists of a set of nodes linked by arcs (or branches). The notation for describing a network is (N, A), where N is the set of nodes, and A is the set of arcs.

What is node?

Often called a vertex, or point, it is normally represented by a circle O. In network, these might be locations, or cities on a map.

What is arc?

Often called an edge, or arrow. It may be either directed or undirected. The head is the destination, the tail is the origin. The head and tail are nodes that are at either end.

$$\underset{\text{Origin}}{A\,O} \longrightarrow \underset{\text{Destination}}{O\,B}$$

A one-way street might be represented by a directed arc. A two-way street might be an undirected arc or by two directed arcs that point in opposite directions. A network with n nodes could have as many as $n!/[(n-2)!2!] = n(n-1)/2$ arcs. If directed, this number might be doubled.

A network consists of a set of nodes linked by arcs (or branches). The notation for describing a network is (N, A), where N is the set of nodes, and A is the set of arcs.

Network Optimization:

The problems of network in which we investigate those values for the dimensions of the links and/or for the flows on the links which optimize the objective function (maximization or minimization), i.e. The technique which apply to the network problems to optimize the objective function under the given circumstances.

For the solution of different situations/problems such as minimum cost, length, max. flow, profit etc. in networks and other like, the following family of network optimization techniques can be used:

1. Minimal Spanning Tree.
2. Shortest route/path algorithm.
3. Maximum flow algorithm.
4. Minimum cost network flow algorithm.
5. Critical Path algorithm (CPM).
6. Assignment, Transportation.
7. Simulation Technique, etc.

1. Minimal Spanning Tree:

Spanning tree is a tree (§section 2.2) that includes every node of the graph. The minimal spanning tree algorithm deals with linking the nodes of a network, directly or indirectly, using the shortest length of connected branches.

This technique can be used in the set of roads, power/telephone supply cables, or gas pipe lines which connect a number of localities with the smallest total distance or cost.

2. Shortest Path Algorithm:

The shortest path algorithm is used to determine the shortest route between a source to destination in a network. Although, it is not always necessary to be concerned with measuring the length of a path; the same methods can be used for paths with costs associated with them.

This can be used in Production planning, Knapsack problem, Businessman tours, Communication network, Replacement problem, Transportation problems, etc.

3. Maximum Flow Algorithm:

In a network with flow capacities on the arcs, the problem is to determine the maximum possible flow from the source to the sink while honoring the arc flow capacities. Thus, the idea of the maximum flow algorithm is to find a breakthrough path with net positive flow that links the source and sink nodes. This technique can be used in different problems such as: A water supply undertaking has a network of supply canals and pipes linking reservoirs, well and pumping stations to its customers, flow of traffic, of goods in a factory, of people and of telephone calls in a network of exchange etc.

4. Minimum Cost Network Flow Algorithm:

The technique can be used to determines the flows in the different arcs that minimize the total cost while satisfying the flows restriction on the arcs and the supply and demand amounts at the nodes. This technique can also be used in the set of nodes, power/telephone supply cables, or a gas pipeline, traveling salesman problems, etc.

5. Critical Path Algorithm:

Critical Path Method, abbreviated as CPM is also a network technique but it is concerned with obtaining the trade-off, between cost and completion date for large projects. CPM is not concerned with uncertain jobs as in PERT (Programme Evaluation and Review Technique— It is one of the Network analysis technique that utilize a network to complete a predetermined project or schedule. it is "event-oriented' emphasizing the descriptions associated with the events). CPM is mostly used in construction projects, or in situation already handled, so that the details like the normal completion time, crash duration and cost of crashing are already known.

6. Assignment, Transportation, etc.

The Assignment problem is also an allocation problem. We have n jobs to perform with n persons and the problem is how to distribute the jobs to the different persons involved. Then the objective function in assigning the different jobs to different persons is to find the optimal assignment that will minimize the total time taken to finish all the jobs by the individuals.

In a Transportation problem, we have certain origins which may represent factories where we produce items and supply a required quantity of the products to a certain number of destinations. This must be done in such a way as to maximize the profit or minimize the cost. Some times the origins and destinations are also termed as sources and sinks.

Traveling Salesman Problem: a large variety of problem, other than the routing one may be developed in connection with the construction and utilization of network. The special type of routing problem, that occurs most frequently in Operations Research – The Traveling Salesman Problem. Suppose a salesman has to visit n cities. He wishes to start from a particular city, visit each city once, and then return to his starting point. The objective is to select the sequence/route in which the cities are visited in such a way that his total traveling time is minimized.

The problem can be represented as a network where the nodes represent the cities and arcs represent the distances between them. The Traveling Salesman Problem is very similar to the assignment problem except that in the former, there is an additional restrictions.

7. Simulation Technique

Simulation modeling can be used both as an analysis tool for predicting the effect of changes to existing systems, and as a design tool to predict the performance of new systems under varying sets of circumstances.

Simulation is not an optimization procedure like linear programming. It allows us to make statement like, "your cost will be C if you take action X", but it does not provide answer like, "Cost is minimized if you take action Y".

This dissertation consists two case studies and related topics.

Chapter-2, discusses two methods for finding the spanning tree for the *Optical Fibre Based Campus Wide Network Phase-I of Aligarh Muslim University (A.M.U.)*, which gives the shortest route through which the network can be established among the various departments.

Chapter-3, discusses three methods [The Simplex Method, A Labeling Algorithm and Out-of-Kilter Algorithm], which solve the minimum cost network flow problem.

Chapter-4, Discusses the Shortest Path Algorithm for the network problems. It is devoted to the discussion of Dijkstra's method and Generalized method (i.e. Ford's and Floyd's Algorithms) for the Shortest Path Problems in the Network. Using the LINGO Software, *the Shortest Path in the Network among the various Departments in AMU Campus* has been obtained.

Chapter-5, discusses four methods for finding the tour (shortest path) in the networks, to solve a *Network Problem of the Distribution of Provision to the Halls/Residents at the AMU Campus.* A brief comparative study of algorithms is also given.

Chapter-6, contains the discussion on the Simulation Technique, for a situation where if the real data is not available, then how to solve such a problem. This Chapter also consists of *A Case Study of Distribution Problem* which is solved by Simulation technique using the generated random numbers, and it is observed that it gives the similar results as obtained in Chapter-5.

Chapter- 7 deals with the conclusion of the dissertation.

Chapter-2

2.1 Introduction:

A real life network problem related to the campus of Aligarh Muslim University is given in section 2.2. This network problem required a solution in the form of a shortest route which has been obtained by the Minimal Spanning Tree.

2.2 Construction of Minimal Spanning Tree:

A network consists of a set of nodes linked by arcs (or branches). The notation for describing a network is (N,A), where N is the set of nodes and A is the set of arcs. e.g. a Network with 5 nodes can be described as:

$$N = \{1,2,3,4,5\}$$
$$A = \{(1,3), (1,2), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5)\}$$

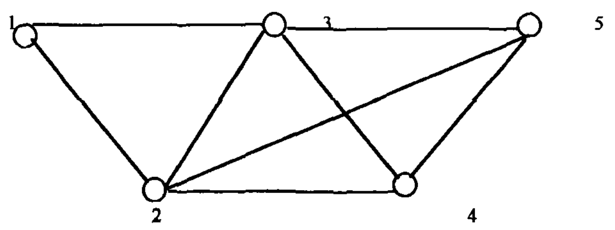


Fig: 2.1

An Arc is said to be directed or oriented if it allows positive flow in one direction and zero flow in the opposite direction. A directed network has all directed branches.

A connected network is such that every two distinct nodes are linked by at least one path. The network above in fig is of this type.

Tree:

A tree is a connected network that may involve only a subset of all the nodes of the network. Hence, it is a connected graph having no cycles. A tree with m nodes has n-1 arcs.

Many human events can be represented by a tree. e.g. family tree, it is also appropriate for representing the way that article are sorted. Letters and parcels in the post are sorted according their destination, and this process is carried out in a number of stages:

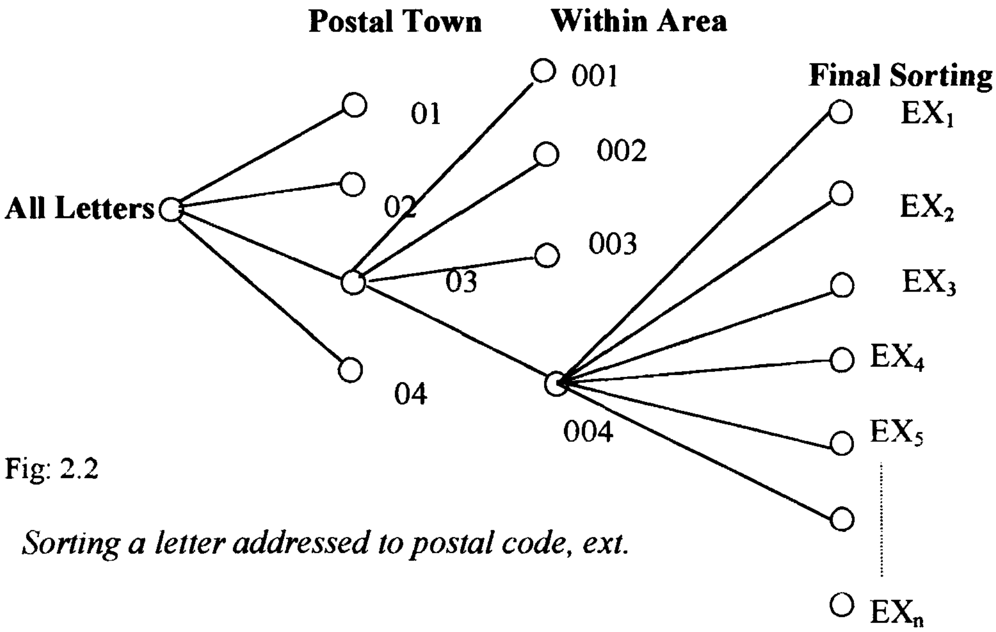


Fig: 2.2

Sorting a letter addressed to postal code, ext.

Minimal Spanning Tree: -

A Spanning tree for the graph (N, A) is a set of arcs, which forms a tree and a partial graph. So, a spanning tree is a tree which contains a path between every pairs of nodes in the graph, and is, in a sense, 'minimal', in that every arc is needed. Removal of any one arc form a Spanning tree means that the graph is no longer 'spanned' in the sense of there being a path between all pairs. The addition of an arc to a spanning tree results in a graph which is not a tree, and which has loops giving alternative paths between some pairs of nodes.

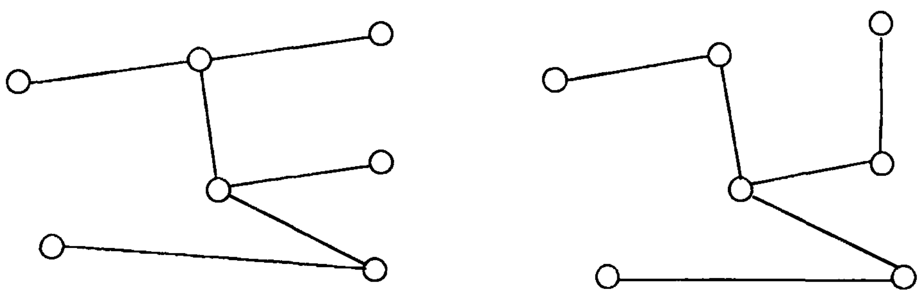


Fig: 2.3

An undirected graph, and one of its spanning tree

When there is a distance measured D associated with the arcs, to give a network (N, A, D) it is possible to define the length of a spanning tree. Thus, different tree may be compared in respect of their lengths. (The distance measured need not to be a physical length of wire/code, it may refer to a cost, or to a fuel consumption, or to a travel time.)

One spanning tree of especial interest is the minimal spanning tree; this is the spanning tree whose length is least, for a given network. The typical application occurs in the creations of a network of paved roads, power/telephone supply cables or gas pipeline etc.

Construction of a Minimal Spanning Tree is a straight forward. It is evident that the arcs in such a tree will be taken from the shortest arcs in the network, in order that a minimal total to be achieved.

There are two algorithms to construct a Minimal Spanning Tree:

(i) Kruskal's Algorithm

(ii) Prim's Algorithm.

2.2.1 Kruskal's Algorithm for a Network

In this method, all the arcs are ranked by length, and are added to the set if they do not form a loop with those arcs which are already in the set; this method is due to J.B. Kruskal, in this, several sub trees being grown simultaneously.

Algorithm: For a network (N, A, D)

Step 0: Initialise the graph T with n nodes and no arcs.

Step 1: Create a list L of arcs from N in ascending order of length, (Rank arbitrarily, those arcs with the same length).

Step 2: Select the arc (i,j) from the head of L. If it forms a circuit in T, delete it from L, and repeat step 2. Otherwise, transfer it from L to T.

Step 3: If T is a tree, stop; otherwise repeat step 2.

2.2.2 Prime Algorithm for a Network:

Prime's method is due to R.C. Prime is sometimes called the 'greedy algorithm'. The spanning tree is grown steadily, from one initial arc; at each stage of the algorithm, the arc which added is the shortest arc remaining which has one vertex in the tree. So, to select an arc, those which are not yet in the partial tree arc scanned, and divided into two sets, corresponding to those which have one (and only one) node in the tree, and those which have none or two. Those with two would form loops, if they were to be added, and those with no nodes could not be added since they would not form a tree. The later cases are discarded, and the shortest arc from the former is added to the partial tree. This adds one further node to the tree, so the sets to be altered, and the process repeated until the partial tree is a spanning tree for the network.

Since, Kruskal's method may create several small trees before the spanning tree is completed, where Prime's method steadily expands one partial tree to become spanning trees.

Algorithm: For a network (N, A, D)

Step 0: Initialise the graph T to have one node, select arbitrarily, and no arcs.

Step 1: Select the arc *(i, j)* whose length is least, from among arcs *(i, l)* which have I in T, l not in T, add this arc to T, and add j to the node set of T.

Step 2: If T is a Spanning tree for the graph (N, A), stop; otherwise, repeat step1.

2.3 A Case Study of Optical Fibre Based Campus Wide Network Phase-I of AMU

The campus of Aligarh Muslim University is spreaded over 467.6 hectares of land (See A.M.U. Campus Map). It comprises of 88 departments of studies. These departments are grouped to form faculties. The departments have varying number of computers. In addition to these, there is a central computing facility known as Computer Centre. The Centre has own Internet connection via VSAT (Very Small Aperture Terminal). Presently the Internet service is available on approximately 80 nodes in the Computer Centre and it is also accessible via dial-up facility. The dial-up facility is extended through 16 telephone lines with hunting facility. To provide access to the server (for computing) and to link directly the Internet service an optical fibre network is being laid down. During its first phase (taken up in 2002-2003) twenty three (23) departments are decided to be connected with the computer centre. In this dissertation the problem of optical fibre network is discussed for 19 departments only which included Computer Centre also (Three departments: Applied Maths, Applied Physics and Applied Chemistry are grouped as one node of Applied Sciences. The departments which were optional in the beginning – Botany, Chemistry, Geology, Zoology and Geography are not included in the problem. The department of Law is also included in the problem while for the phase–I it is excluded). The distances among the departments are given in Table –1 in distance square matrix form. Now the problem under study is to find a minimal spanning tree for the optical fibre network of campus wide networking phase-I. For the above described situation an "Integer Programming Problem" is formulated in LINGO software (see Appendix C). This problem is solved in two parts of network. At node 12 (Deptt. of Civil Engineering) there is a junction from which further network is extended to other neighbouring departments. So, first part of the network is a set of nodes 1 – 11 and the second set of nodes is 12-19. A general network graph which connects each concerned departments different routes is shown in Fig. 2.4.

Discussion for the Solution of the Problem:

The problem under study is to find a minimal spanning tree for the optical fibre network of campus wide networking phase – I. Given a set of nodes (Department) to be connected, we want to find a minimum cost network, so every node is connected to the network. A reasonable approximation to the cost of network is the sum of the cost of arcs in the network.

In fact, formulation of the minimum spanning tree problem as Linear Programming is a bit tedious, so, the LINGO Model for spanning tree is used to solve it as an Integer Program. The corresponding values of X in the solution of spanning tree are given in the Appendix –C. The value of X=1 in the solution allow the tree to form spanning tree.

First part of the Spanning Tree of a Network is:

1 → 7 → 9 → 8

1 → 2 → 3 → 4 → 6 → 5

4 → 10 → 11

Graph is given in Fig- 2.5

Second part of spanning Tree of a Network is:

12 → 13 → 14 → 15 → 16 → 17

13 → 18 → 19

Graph is given in Fig – 2.6

The total Optimum length of optical fibre required under this network (Network-I and Network-II) is:

= 1364.00 meters + 489.500 meters

= 1853.5 meters

Since cost of optical fibre cable per meter is = Rs. 80/-

Hence the total cost required to set up the network of phase-I = 1853*80

= Rs. 1,48240/-

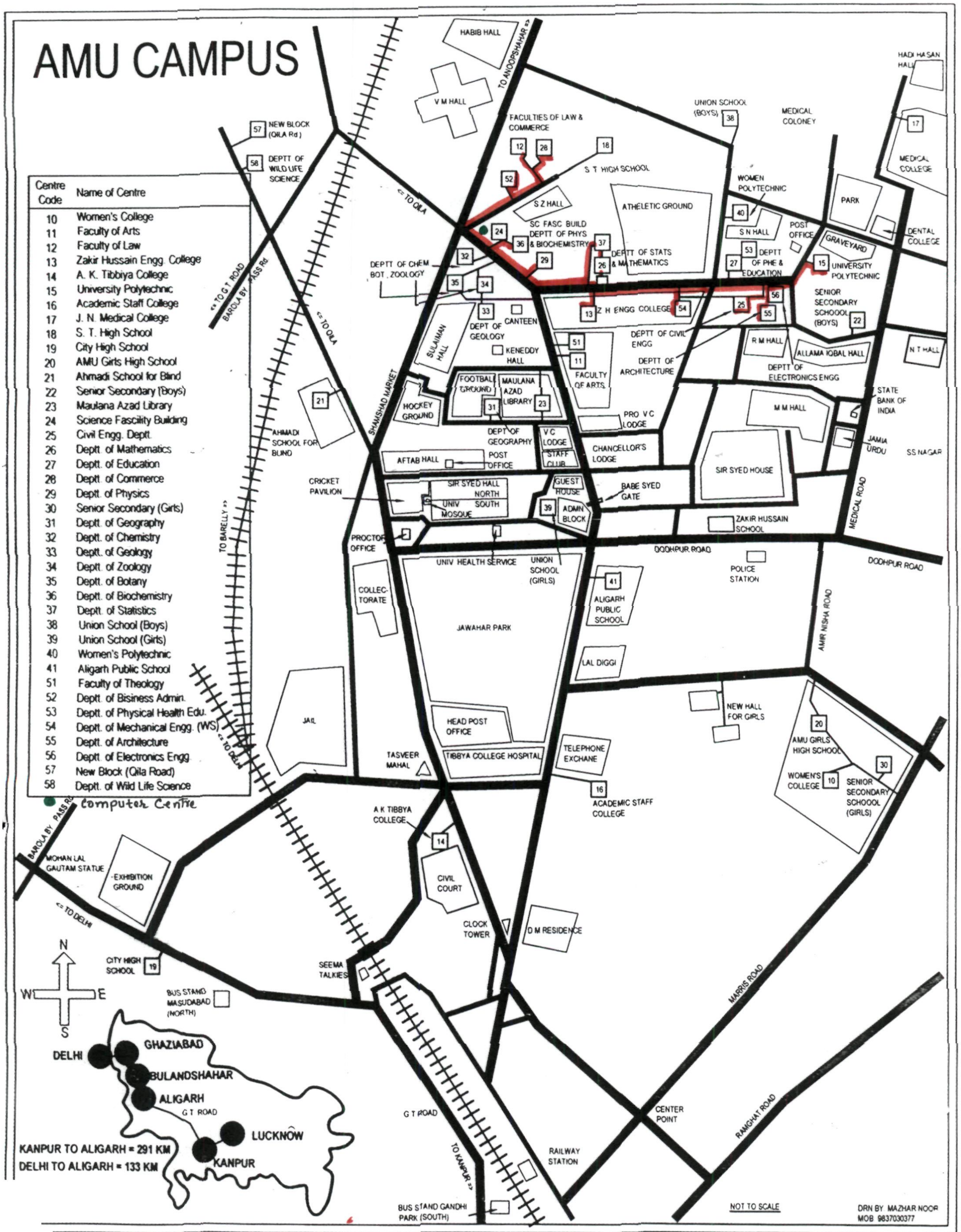
DATA SET

Table-1

Distance matrix (in meters)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	D TO D	CC	BIOCH	PHY	CS	STATS	MATHS	LAW	MBA	COMM	MED	MEDW	CED	AS	ELCAL	ELNIC	COMP	UPOLY	BARCH	CHE/PAT
1	CC	0.0	45.7	207.3	205.7	442.0	391.1	435.0	390.0	490.0	411.5	527.9	640.1	640.1	716.3	771.2	771.2	876.3	883.9	713.2
2	BIOCH	45.7	0.0	50.0	251.4	571.4	520.5	480.7	435.2	535.7	541.0	670.6	792.5	792.5	868.7	923.6	929.7	1089.7	929.7	957.1
3	PHY	207.7	50.0	0.0	50.0	370.0	319.1	642.3	597.3	997.3	349.6	456.0	597.9	577.9	654.1	709.0	715.1	875.1	715.1	742.5
4	CS	205.7	251.4	50.0	0.0	320.0	269.1	640.7	595.7	695.7	289.6	406.0	518.2	518.2	594.2	649.3	655.4	815.4	655.4	682.8
5	STATS	442.0	571.4	370.0	320.0	0.0	30.0	877.0	832.0	932.0	609.6	726.0	847.9	847.9	924.1	979.0	985.1	1145.1	985.1	1012.5
6	MATHS	391.1	520.5	319.1	269.1	30.0	0.0	826.1	781.1	881.1	558.7	675.1	797.0	797.0	873.2	928.1	934.2	1094.2	934.2	961.6
7	LAW	435.0	480.7	642.3	640.7	960.7	826.1	0.0	110.0	30.0	846.5	962.9	1075.1	1075.1	1151.3	1206.2	1206.2	1311.3	1318.9	1148.2
8	MBA	390.0	435.7	597.3	595.7	915.7	781.1	110.0	0.0	80.0	801.5	917.9	1030.1	1030.1	1106.3	1161.2	1161.2	1266.3	1273.9	1103.2
9	COMM	490.0	535.7	697.3	695.7	932.0	881.1	30.0	80.0	0.0	901.5	1017.9	1130.1	1130.1	1206.3	1261.2	1261.2	1366.3	1373.9	1203.2
10	MED	411.5	541.0	339.6	289.6	609.6	558.7	846.5	801.5	901.5	0.0	129.6	251.5	251.5	327.7	382.6	388.7	548.7	388.7	416.1
11	MEDW	527.9	670.6	456.0	406.0	726.0	675.1	962.9	917.5	1017.9	129.6	0.0	121.9	121.9	198.1	253.0	259.1	419.1	259.1	286.5
12	CED	640.1	792.5	577.9	518.2	847.9	797.0	1075.1	1030.1	1130.1	251.5	121.9	0.0	0.0	76.2	131.1	137.2	297.2	137.2	164.6
13	AS	640.1	792.5	577.9	518.2	847.9	797.0	1075.1	1030.1	1130.1	251.5	121.9	0.0	0.0	76.2	131.1	137.2	297.2	137.2	164.6
14	ELCAL	716.3	868.7	654.1	594.4	924.1	873.2	1151.3	1106.3	1206.3	327.2	198.1	76.2	76.2	0.0	50.0	56.1	216.1	213.4	240.8
15	ELNIC	771.2	923.6	709.0	649.3	979.0	928.1	1206.2	1161.2	1261.2	382.6	253.0	131.1	131.1	50.0	0.0	6.1	166.1	268.3	295.7
16	COMP	771.2	929.7	715.1	655.4	985.1	934.2	1206.2	1161.2	1261.2	388.7	259.1	137.2	137.2	56.1	6.1	0.0	160.0	274.4	301.8
17	UPOLY	876.3	1089.7	875.1	815.4	1145.1	1094.2	1311.3	1266.3	1366.3	548.7	419.1	297.2	297.2	216.1	166.1	160.0	0.0	434.4	461.8
18	BARCH	883.9	929.7	715.1	655.4	985.1	934.2	1318.9	1273.3	1373.9	388.7	259.1	137.2	137.2	213.4	268.3	274.4	434.4	0.0	60.0
19	CHE/PAT	713.2	957.1	742.5	682.8	1012.5	961.6	1148.2	1103.2	1203.2	416.1	286.5	164.6	164.6	240.8	295.7	301.8	461.8	60.0	0.0

AMU CAMPUS



Centre Code	Name of Centre
10	Women's College
11	Faculty of Arts
12	Faculty of Law
13	Zakir Hussain Engg. College
14	A. K. Tibbiya College
15	University Polytechnic
16	Academic Staff College
17	J. N. Medical College
18	S. T. High School
19	City High School
20	AMU Girls High School
21	Ahmadi School for Blind
22	Senior Secondary (Boys)
23	Maulana Azad Library
24	Science Fascility Building
25	Civil Engg. Deptt.
26	Deptt. of Mathematics
27	Deptt. of Education
28	Deptt. of Commerce
29	Deptt. of Physics
30	Senior Secondary (Girls)
31	Deptt. of Geography
32	Deptt. of Chemistry
33	Deptt. of Geology
34	Deptt. of Zoology
35	Deptt. of Botany
36	Deptt. of Biochemistry
37	Deptt. of Statistics
38	Union School (Boys)
39	Union School (Girls)
40	Women's Polytechnic
41	Aligarh Public School
51	Faculty of Theology
52	Deptt. of Bisiness Admin.
53	Deptt. of Physical Health Edu.
54	Deptt. of Mechanical Engg. (WS)
55	Deptt. of Architecture
56	Deptt. of Electronics Engg.
57	New Block (Qila Road)
58	Deptt. of Wild Life Science

Computer Centre

AMU CAMPUS NETWORK

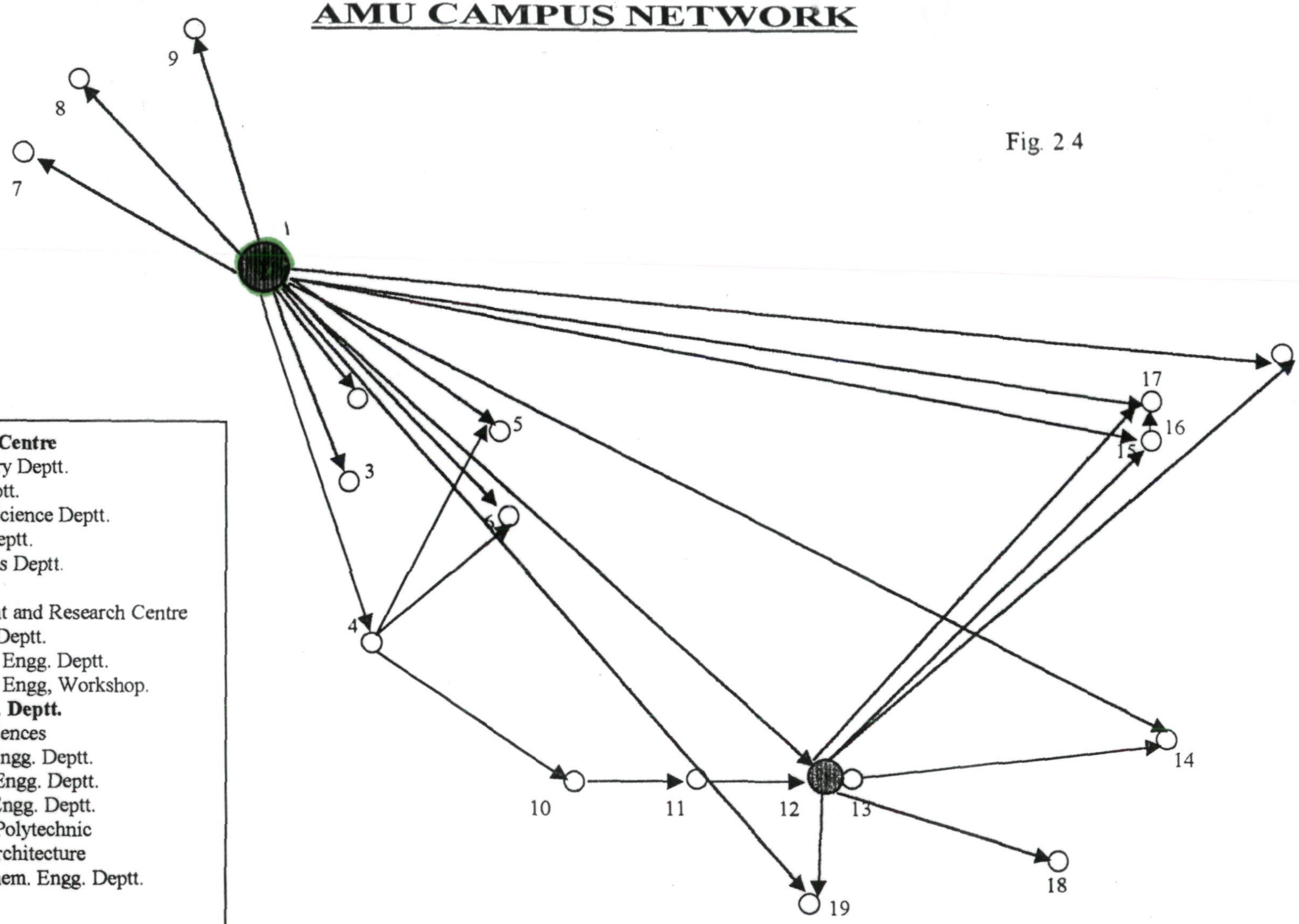
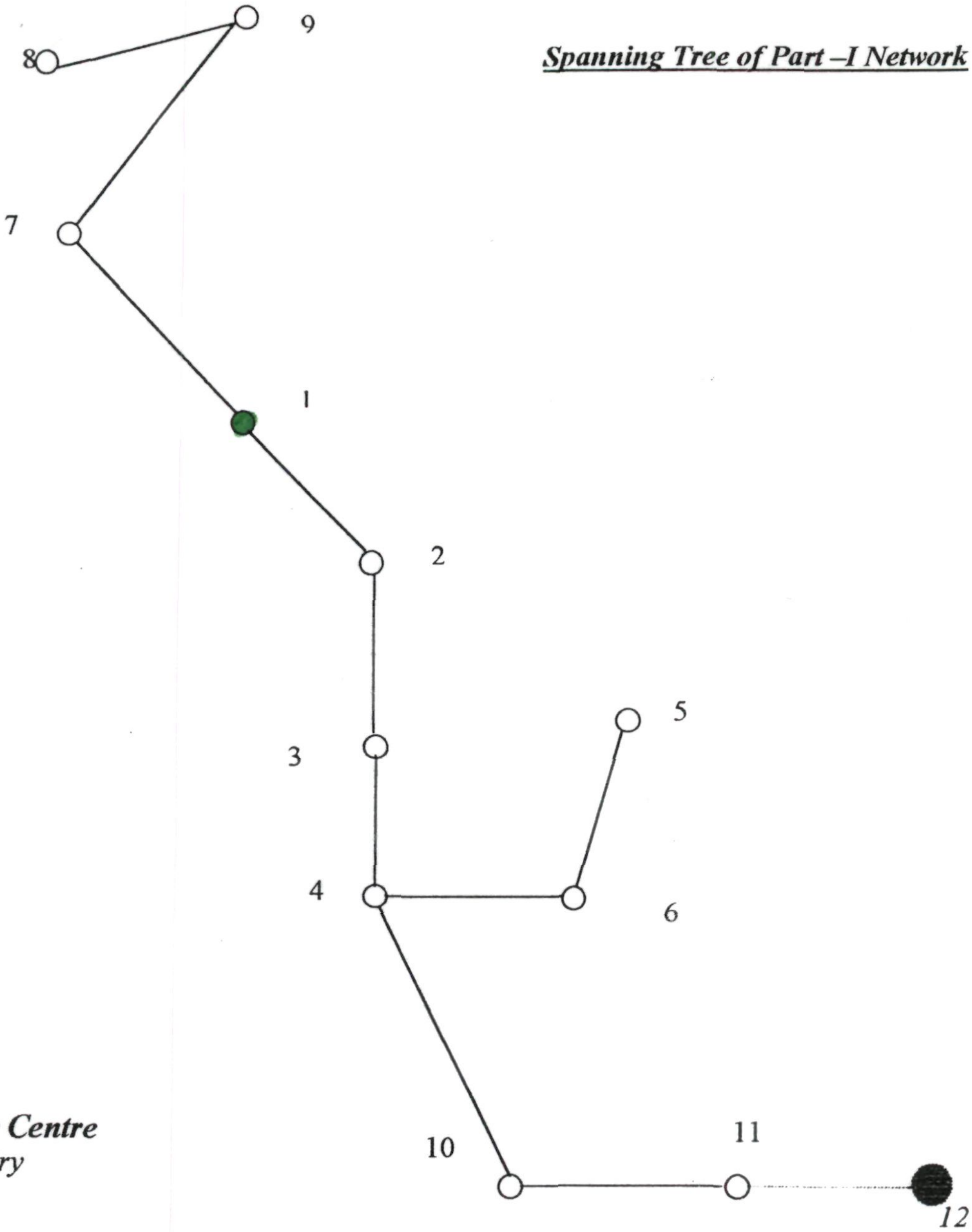


Fig. 2 4

1- **Computer Centre**
2- Biochemistry Deptt.
3- Physics Deptt.
4- Computer Science Deptt.
5- Statistics Deptt.
6- Mathematics Deptt.
7- Law Deptt.
8- Management and Research Centre
9- Commerce Deptt.
10- Mechanical Engg. Deptt.
11- Mechanical Engg, Workshop.
12- Civil Engg. Deptt.
13- Applied Sciences
14- Electrical Engg. Deptt.
15- Electronic Engg. Deptt.
16- Computer Engg. Deptt.
17- University Polytechnic
18- Deptt. of Architecture
19- Petro. & Chem. Engg. Deptt.

Corresponding values of X to the following spanning tree are given in APPENDIX – C.

Spanning Tree of Part –I Network



1- ***Computer Centre***
2- Biochemistry
3- Physics
4- Comp. Science
5- Stats Deptt.
6- Maths Deptt .
7- Mangt. & Research
8- Deptt. of Law
9- Deptt .of Commerce
10- Mechanical Engg. Deptt.
11- Mechanical Engg. Workshop.

Fig. 2.5

Corresponding values of X to the following spanning tree are given in APPENDIX – C.

Spanning Tree of Part -II

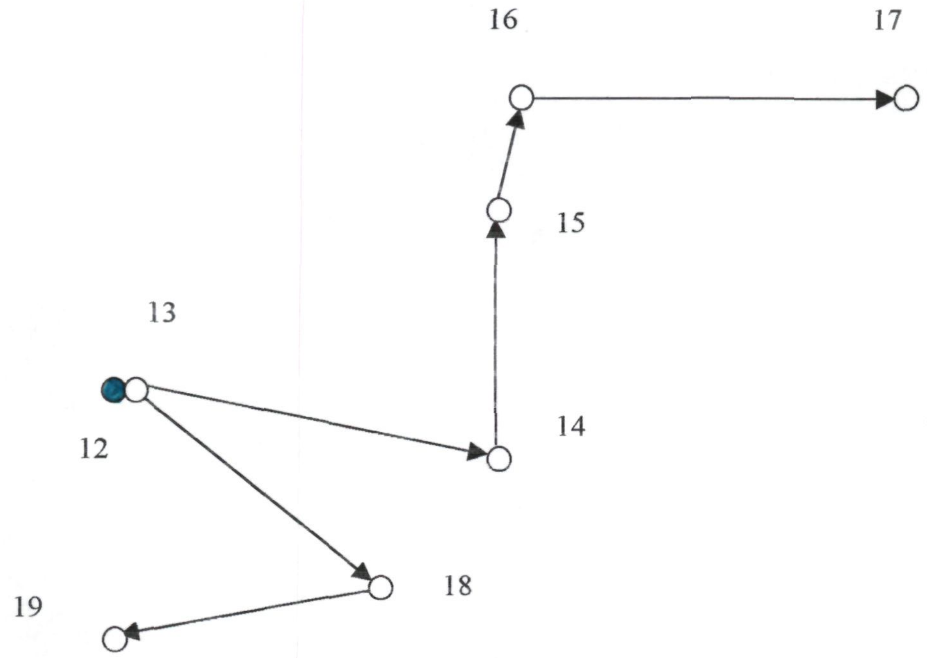


Fig. 2.6

12 – Civil Engg. Deptt.

13 – Applied Science

14 –Electrical Engg. Deptt.

15 –Electronic Engg. Deptt.

16 – Computer Engg. Deptt.

17 –Univ. Polytechnic

18 – Deptt. of Architecture

19 – Deptt. of Chemical/Petrolium Engg.

Chapter-3

Minimum Cost Network Flow Chapter – 3

3.1 Introduction:

A general problem is that of finding a flow in a network which has minimal total cost, subject to upper and lower limits on the flows in individual arcs. This problem is known as the minimal – cost – flow problem.

This problem may be regarded as a problem of sending one unit of flow from one city to another at minimal cost (distance). In the general minimal-cost flow problem, there are no source and sink.

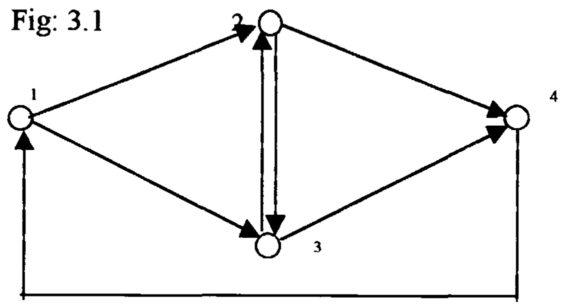
The minimal-cost-flow problem is a linear programming problem. It has an objective function which is linear, and all the constraints both on flow size and on flow conservation, are linear equality constraints when slacks variables have been introduced. This problem has a very simple form, as the constraints are of three types only, corresponding to:

- Flow conservation at nodes.
- lower bounds on flows.
- upper bounds on flows.

The minimal cost flow problem might arise in a logistic network where men and materials are being moved between various points in the world. It may be associated with the movement of locomotives between points in a rail road network to satisfy power for train at least travel cost.

Consider a *directed network* G, consisting of a finite set of *nodes* (points) N = {1,2,3, ………..m} and a set of *directed arcs* (lines) $S = \{(i,j),\ (k,i),\ \ldots\ldots\ldots\ldots..(s,t)\}$ joining pairs of nodes in N. Arc (i,j) is said to be *incident* with node i and j and is directed from node i to node j. We shall assume that the network has m nodes and n arcs (in fig.3.1, 4 nodes & 7 nodes).

Fig: 3.1



With each node i in G we associate a number b_i that is the available supply of an item (if $b_i > 0$) or the required demand for the item (if $bi<o$). Nodes with $b_i > 0$ are sometimes called sources, and nodes with $b_i<0$ are sometimes called sinks. If $b_i=0$, then none of them item is available at node I and none is required; in this case node I is sometimes called an intermediate (or transhipment) node. Associated with each arc *(i,j)* we let x_{ij} be the amount of flow on the arc (we assume $0 \leq x_{ij}$) and c_{ij} be the unit shipping cost along the arc.

The minimum cost network flow problem may be stated as: Ship the available supply through the network to satisfy demand at minimum cost.
Mathematically, this problem is as

$$\begin{aligned} &Minimize \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij} \\ &Subject\ to \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = b_i \quad ,i = 1,2,3,\ldots\ldots\ldots m \qquad (1) \\ &\qquad\qquad x_{ij} \geq 0, \quad i, j = 1,2,\ldots\ldots\ldots\ldots m \end{aligned}$$

Constraints (1) are called the *flow conservation* or *Kirchhoff* equations and indicate that the flow may be neither created nor destroyed in the network. In the conservation equations, $\sum_{j=1}^{m} x_{ij}$ represents the total flow out of node i while $\sum_{k=1}^{m} x_{ki}$ indicates the total flow into node i. These equations required that net flow out of node I, $\sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki}$, should equal b_i. If $b_i < 0$, then there should be more flow into i than out of i.

Minimum cost network flow problems occur in the design and analysis of communication system, oil pipeline systems, tanker scheduling problems, and a variety of other areas.

Clearly, this problem is a linear programme and can be solved in any one of several ways. One way is to apply the ordinary primal simplex algorithm to the problem.

3.2 THE SIMPLEX METHOD FOR NETWORK FLOW PROBLEM

The MCFP can be converted to an equivalent transportation problem with m sources and m destination centres. But, Simplex method can be used on the network itself without converting the problem to an equivalent transportation problem.

The general steps of the simplex method are as: First find a starting basic feasible solution. Next, compute z_j-c_i for each non-basic variable x_j if optimality is achieved, stop, otherwise select the entering column. If optimality is not achieved, determine the exit (blocking) column and pivot.

1. Finding the initial basic feasible solution:-

Consider the A matrix without the additional artificial column. Suppose that we add an artificial column for every row of A, the ith artificial column being $\pm e$, depending on the sign of b_i (i.e. $\pm e_i , b_i \geq 0, - e_i$ otherwise). Also, let us add a redundant row given by the negative of the sum of the rows of the "extended" A matrix.
The problem then becomes:

$$\left[\begin{array}{c|ccccc} & \pm 1 & & & & \\ & & \pm 1 & & & \\ A & & & \pm 1 & & \\ & & & & \ddots & \\ & & & & & \pm 1 \\ \hline 0 & \pm 1 & \pm 1 & \pm 1 & \cdots & \pm 1 \end{array}\right] \qquad \left[\begin{array}{c} \\ \\ b \\ \\ \\ \hline 0 \end{array}\right]$$

Since each column of this "new" A matrix has exactly one +1 and one −1, we may view it as a node-arc incidence matrix of a graph. This "new" graph has all of the same nodes and arcs as the original graph. In addition, it has a new node and m new arcs-one arc between each original node and the new node. A feasible basis for this problem is given by the m artificial variables (arcs) plus an additional artificial variable (root for the new row $(m+1)$).

Beginning with the artificial basis, we may proceed to apply the two-phase method or the big-M method, using approximate costs in each case, until feasibility is achieved. At that time, we may drop all of the artificial arcs (variables) and node $(m+1)$, and replace these by a single artificial variable (root) at node m.

2. Computing the values of the Basic Variables:-

Basic variables can be obtained by two ways, as either by adding the artificial arc and taking advantage of the lower triangular structure of the basic matrix, and may iteratively solve for the basic variables or computing directly on the graph. The process of obtaining the basic solution proceedings from the end of the tree towards the roots.

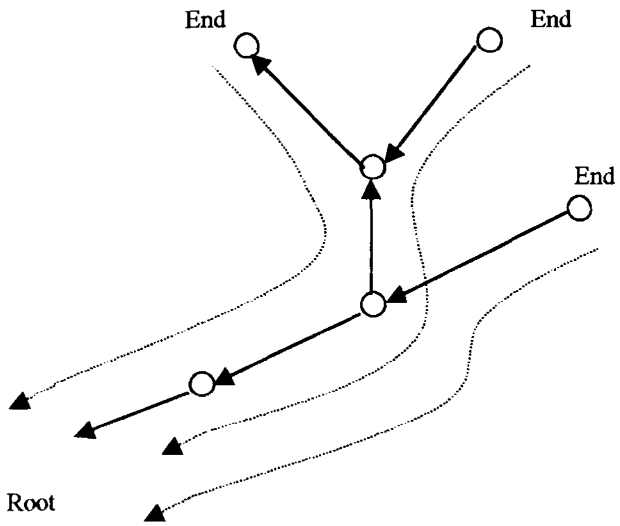


Fig: 3.2

Computing the values of the basic variables.

3. Computing the Z_{ij} - C_{ij}:-

For a given sub-graph, compute z_{ij} - c_{ij} for each non-basic variables x_{ij} and either stop or proceed by introducing a non-basic variable with a positive z_{ij} - c_{ij}.

An easy method of computing z_{ij} - c_{ij} for a non basic arc is to compute the dual vector, w, and determine z_{ij} - c_{ij} through the expression $z_{ij} - c_{ij} = w_{aij} - c_{ij}$. In order to compute w, the system $w_B\, c_B$ must be solved.

We start with the dual variable for the root of node at zero value, then proceed away from the root toward the ends of the tree using the relationship that $w_i - w_j = c_{ij}$ along the basic arc in the tree.

To compute the $z_{ij} - c_{ij}$ for the non-basic arc (ij), we apply the definition:

$$\begin{aligned} z_{ij} - c_{ij} &= w_{aij} - c_{ij} \\ &= w(e_i - e_j) - c_{ij} \\ &= w_i - w_j - c_{ij} \end{aligned}$$

Thus, the $z_{ij} - c_{ij}$ can be conveniently computed on the network.

Note: While the process of computing primal variables consisted of working from the ends of the basis tree inward towards the root, the process of computing dual variables consists of working from the root of the basis tree outward toward the ends.

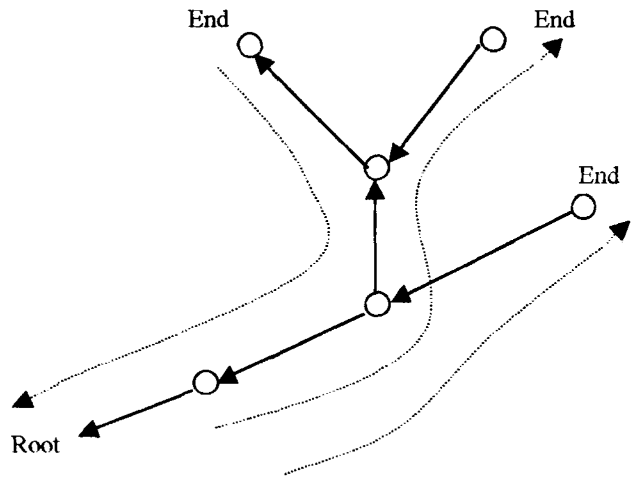
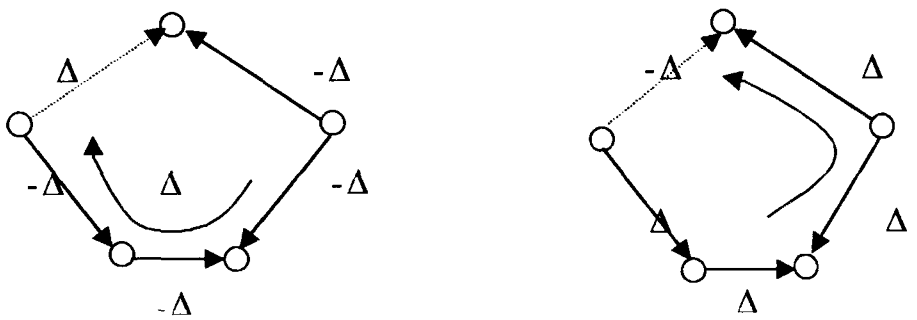


Fig: 3.3 *Computing the values of dual variable.*

4. Determining the Exit Column and Pivoting:-

Once the entering column is selected, it is again an easy task to select the exit column and pivot. We add the entering non-basic arc,, regardless of whether the variable is increasing or decreasing to the basis tree and determine the unique cycle formed. Thus, if the entering variable is increasing, we send an amount Δ around the cycle in the direction of the entering variable, if the entering variable decreasing, we send an amount Δ around the cycle against the direction of the entering variable.



(a) x_{ij} increasing Fig:3.4 (b) x_{ij} decreasing

3.3 Network Flows with Lower and Upper bounds:

It is a simple and straight forward to make the transition from the ordinary simplex method for network flow problems to the lower-upper bound simplex method for network flow problem. There are little changes in simplex method to solve network flow problem.

Getting started:

In this case, since all "real" arcs start out non-basic during phase I (or at the outset of the big-M method). We set all of these arcs below variables at one or other of their bounds and compute the effect on the b values according to

$$\hat{b}_i = b_i - \sum_j x_{ij} + \sum_k x_{ki}$$

Using the vector $\hat{b}$, we establish the "direction" (sign) of the artificial columns to add and begin phase I.

Computing the values of the basic variables:

Whether in phase I or phase II, or during the big-M procedure, after adjusting the b vector to $\hat{b}$ to reflect the values of the non-basic variables (arcs) are proceed in the same manner as in case of Simplex method to compute the values of the basic flow variables.

Computing the Dual Variables and the Z_{ij}-C_{ij}'s:

Lower and upper bounds have no effect on the competition of the variables and on the computation of the Z_{ij}-C_{ij}'s. However, in presence of lower and upper bound, the optimality criteria are

$x_{ij} = u_{ij} \Longrightarrow z_{ij} - c_{ij} \leq o$

$and\ x_{ij} = l_{ij} \Rightarrow z_{ij} - c_{ij} \leq o$

These are easy to check and we can readily determine whether some non-basic variable x_{ij} should be increased or decreased if optimality is not achieved.

3.4 Labelling Algorithm for the Network Simplex Method:

For either hand or computer calculations there are simple and convenient ways to maintain the information required to solve a minimal cost flow problem with lower and upper bounds by the network simplex method. Suppose that we associated with each not $j \in$ N a level, $L(j) = (\pm i, \Delta_j)$, containing tow pieces of information. The second entry, Δ_j, in $L(j)$ indicates the current estimates for the value of the flow change. The first entry, $\pm i$, in $L(j)$ indicates the previous node in the cycle along which flow will be changed. If the first entry in $L(j)$ is $+i$, then flow will be added to arc (i, j); otherwise, if the first entry is $-i$, then flow will be subtracted from arc (j, i). The labeling algorithm becomes the following:

Initialising Step

Select a basic feasible solution and set the x_{ij}'s to their required values in the solution. If a basic feasible solution is not readily available, utilize artificial variables.

Main Step

1. Set $w_m = 0$. If w_i has been computed, w_j has not been computed, and arc (i, j) is a basic arc, then set $w_j = w_i - c_{ij}$. If w_i has been computer, w_j has not been computed, and are (j,i) is a basic arc, then set $w_j = w_i + c_{ji}$. Repeat step 1 until all w_i's have been computed.

2. If each nonbasic variable has $x_{ij} = l_{ij}$ and $z_{ij} - c_{ij} \leq 0$ or $x_{ij} = u_{ij}$ and $z_{ij} - c_{ij} \geq 0$, stop; the optimal solution is obtained. Otherwise, erase any labels. If $z_{pq} - c_{pq}$ <0 and $x_{pq} = u_{pq}$, set $s = p$, $t = q$, $(g, h) = (p, q)$, and $L(s) = (-t, x_{pq} - l_{pq})$; or if $z_{pq} - c_{pq} > 0$ and $x_{pq} = l_{pq}$, set $s = q$, $t = p$, $(g, h) = (p, q)$ and $L(s) = (+t, u_{pq} - x_{pq})$.

3a. If node i has a label, node j has no label, and arc (i, j) is basic, set $L(j) = (+i, \Delta_j)$ where Δ_j = Minimum $\{\Delta_i, u_{ij} - x_{ij}\}$. If $u_{ij} - x_{ij} < \Delta_i$, set $(g, h) = (i, j)$.

b. If node i has a label, node j has no label, and arc (j, i) is basic, set $L(j) = (-i, \Delta_j)$ where Δ_j = Minimum $\{\Delta i, x_{ij} - l_{ij}\}$. If $x_{ij} - l_{ij} < \Delta_i$, set $(g, h) = (j, i)$.

c. Repeat step 3 until node t is labeled.

4. Let $\Delta = \Delta_t$. If the first entry in $L(t)$ is $+k$, then add Δ to x_{kt}; otherwise, if the first entry in $L(t)$ is $-k$, subtract Δ from x_{lt}. Backtrack to node k and repeat the process until node t is reached in the backtrack process.

5. If $(g, h) = (p, q)$, go to step 2. Otherwise add (p, q) to the basis, remove (g, h) from the basis, and go to step 1.

3.5 Out-of-Kilter Algorithm

Out-of-Kilter algorithm is another method for solving minimal cost network flow problems. This algorithm is similar to the primal-dual algorithm in that it begins with dual feasibility but not necessarily primal feasibility and iterates between primal and dual problems until optimality is achieved. However, it differs from the primal-dual algorithm in that the Out-of-Kilter algorithm does not always maintain complementary slackness. Thus, it can be viewed as a generalisation of the primal-dual algorithm for network flow problem.

The Out-of-Kilter Formulation of Minimal Cost Network Flow Problem

Consider the minimal cost flow problem (MCFP):

$$\begin{array}{llr} \textit{Minimze} & \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} & \\ \textit{Subject to} & \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = 0\ , & i = 1,2,\ldots\ldots\ldots\ldots m.\ (i) \\ & x_{ij} \geq l_{ij}, & i,j = 1,2,\ldots\ldots\ldots\ldots m \\ & x_{ij} \leq l_{ij}, & i,j = 1,2,\ldots\ldots\ldots\ldots m \end{array}$$

Any flow (choice of the x_{ij}'s) which satisfy the constraints (1) is called a conserving flow. A conserving flow that satisfy the remaining constraints $l_{ij} \le x_{ij} \le u_{ij}$ is a feasible flow (solution), we shall assume that c_{ij}, l_{ij} and u_{ij} are integers and that $0 \le l_{ij} \le u_{ij}$.

In general, it may be difficult to obtain a conserving flow which satisfy all the constraints. The Out-of-kilter algorithm uses a conserving flow together with a dual feasible solution which can also be easily obtained.

The Dual of a Network Flow Problem

If we associate a dual variable w_i with each node conservation equation (1), a dual variable h_{ij} with the constraint $x_{ij} \le u_{ij}$ (which is treated as $-x_{ij} \ge -u_{ij}$ for taking dual) and a dual variable v_{ij} with the constraint $x_{ij} \ge l_{ij}$, the dual of the Out-of-Kilter formulation for MCFP is as:

$$\begin{array}{lll} \text{Minimize} & \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{m} l_{ij} v_{ij} - \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{m} u_{ij} h_{ij} & \\ \text{Subject to} & w_i - w_j - v_{ij} - h_{ij} = c_{ij,} & i, j = 1,2,\ldots\ldots \ldots\ldots\ldots\ldots ,m \\ & h_{ij}, v_{ij} \ge 0, & i, j = 1,2\ldots \ldots\ldots\ldots \ldots\ldots, m \\ & w_i \;\; \text{Undirected} \;\; , & i = 1,2,\ldots\ldots \ldots\ldots\ldots \ldots\ldots, m \end{array}$$

where summation and the constraints are taken over existing arcs.

Suppose that we select any set of w_i's, then the dual constraint for arc (i,j) become

$$v_{ij} - h_{ij} = c_{ij} - w_i + w_j, \qquad h_{ij} \ge 0$$
$$v_{ij} \ge 0.$$

and can be satisfied by

$$v_{ij} = Maximum\{0, c_{ij} - w_i + w_j\}$$
$$h_{ij} = Maximum\{0, -(c_{ij} - w_i + w_j)\}$$

Thus, the dual problem always possesses a feasible solution given a set of w_i's.

The Complementary Slackness Conditions

The Complementary slackness conditions for optimality of the Out-of-Kilter formulation are as:

$$\left(x_{ij} - l_{ij}\right) v_{ij} = 0, \qquad i,j = 1,2,\ldots\ldots\ldots\ldots m \qquad (2)$$

$$\left(u_{ij} - x_{ij}\right) h_{ij} = 0, \qquad i,j = 1,2,\ldots\ldots\ldots\ldots m \qquad (3)$$

Define $Z_{ij} - C_{ij} \equiv w_i - w_j - c_{ij}$, then by the definition of v_{ij} and h_{ij}, we get

v_{ij} = Maximum $\{0, -(Z_{ij}-C_{ij})\}$ (5)

h_{ij} = Maximum $\{0, Z_{ij}-C_{ij}\}$ (6)

Given a set of w_i's, we can compute $Z_{ij}-C_{ij} - w_i-w_j-c_{ij}$. Noting equations (4) and (5), then, the complementary slackness conditions (2) & (3) hold only if

$$Z_{ij} - C_{ij} < 0 \Rightarrow v_{ij} > 0 \Rightarrow x_{ij} = l_{ij}, \qquad i,j = 1,2,\ldots\ldots\ldots\ldots m$$

$$Z_{ij} - C_{ij} > 0h \Rightarrow h_{ij} > 0 \Rightarrow x_{ij} = u_{ij}, \qquad i,j = 1,2,\ldots\ldots\ldots\ldots m$$

We include the addtional condition

$$Z_{ij}-C_{ij} = 0 \Rightarrow l_{ij} \le x_{ij} \le u_{ij}, \qquad i,j = 1,2,\ldots\ldots\ldots\ldots m$$

Thus any conserving flow that satisfies the three conditions above will be optimal solution.

The Kilter States for an Arc

There are two states of an arc whether it is in-kilter or out-of-kilter. An arc in a network is in Kilter if $l_{ij} \le x_{ij} \le u_{ij}$.

and conditions (2) and (3) hold. As we change the flow on arc *(i,j)*, the arc moves up and power a particular column depending on whether x_{ij} is increased or decreased. The states of arcs of a network is shown in fig. 3.5

	$Z_{ij}-C_{ij}<0$	$Z_{ij}-C_{ij}=0$	$Z_{ij}-C_{ij}>0$
$x_{ij}>u_{ij}$	Out-of-Kilter	O-K	O-K
$x_{ij}=u_{ij}$	"	In-Kilter	I-K
$l_{ij}<x_{ij}<u_{ij}$	O-K	I-K	O-K
$x_{ij}=l_{ij}$	I-K	I-K	O-K
$x_{ij}<l_{ij}$	O-K	O-K	O-K

(a)

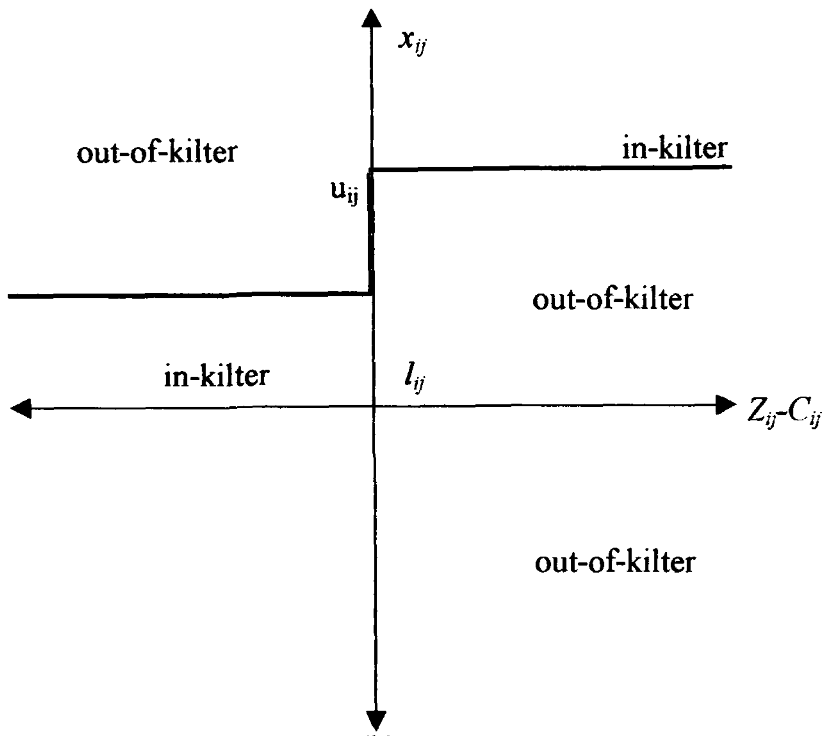


Fig: 3.5

(b)
The possible Kilter states for an arc.

Kilter Number

There are many different measures of distance for the Out-of-Kilter problem. One measure of distance that is called the Kilter Number K_{ij} for an arc (i,j).
The Kilter number is defined as to be the minimal change of flow on the arc that is needed to bring it into kilter, The Kilter number for an arc u_i*'s* is non-negative and if the arc is in-kilter, the associated kilter number is zero, and it is strictly positive if an arc is out-of-kilter.

Strategy of the out-of-Kilter Algorithm

Step-1: Begin with a conserving flow, such as each x_{ij}=*0*, and a feasible solution to the dual , such as each w_i = *0*, with h_{ij}, v_{ij}, identify the kilter states and compute the kilter numbers.

Step-2: If the network has an out-of-kilter arc, conduct a primal phase of the algorithm. During this phase an out-of-kilter arc is selected and an attempt is made to construct a new conserving flow in such a way that a kilter number of no arc is worsened and that of the selected arc is improved.

Step-3: When it is determined that no such improving flow can be constructed during the primal phase, the algorithm constructs a new dual solution in such a way that no-kilter number is worsened and step-2 is repeated.

Step-4: Iterating between step-2 and 3, the algorithm eventually constructs an optimal solution or determines that no feasible solution exists.

3.6 Multi-Commodity Minimum Cost Flow Problem

Introduction:

There is another class of network flow problems called multi-commodity flow problems in which it is necessary to distinguish among the flows in the network. But this flows do not enjoy the same special properties as single-commodity flow problems and also do not necessarily provide integer flows.

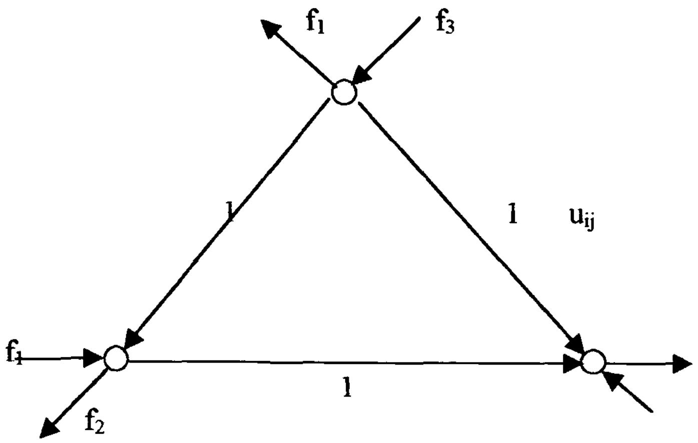


Fig: 3.6 *A three-commodity maximal flow problem.*

Suppose that three commodities that flow through the network . The source for commodity 1 is node 1, and sink is node 3. Similarly, let the source and sink for commodity 2 be nodes 2 and 1 respectively, and finally, the source and sink for commodity 3 are nodes 3 and 2 respectively with the sum of all commodities flowing on an arc should not exceed the arc capacity, $u_{ij} = 1$.

Formulation

Suppose that we are given a network G with m nodes and n arcs in which there will flow t different commodities. Let u_i represent the vector of upper limits on flow for commodity I in the arcs of the network. Then u_{ipq} is the upper limit on flow of commodity I in arc (p,q). Also, let u represents the vector of upper limits on the sum of all commodities flowing in the arcs of the network. Then u_{pq} is the upper limit on the sum of all commodity flows in arc (p, q). Let C_i represents the vector of arc costs in the network for commodity i. Then C_{ipq} is the unit cost of commodity I on arc (p,q). finally, let b_i represents the vector of supplies (or demands) of commodity I in the network. Then b_{iq} is the supply (if $b_{iq}>0$) of demand (if $b_{iq}<0$) of commodity I at node q.

Thus, the linear programming formulation for the Multi-commodity minimal cost flow problem is as follows:

$$
\begin{aligned}
&\textit{Minimize} \sum_{i=1}^{t} c_i x_i \\
&\textit{Subject to} \sum_{i=1}^{t} x_i \le u \\
&\qquad\qquad\qquad Ax_i = b_i, i = 1,2,\dots\dots\dots\dots,t \\
&\qquad\qquad\qquad 0 \le x_i \le u_i, i = 1,2,\dots\dots\dots\dots,t
\end{aligned}
$$

where x_i is the vector of flows of commodity I in the network and A is the node-arc incidence matrix of the graph. Since, this problem possesses the block diagonal structure.

Thus, we may apply the block diagonal decomposition technique to solve the above problem.

The Multi-commodity minimal cost flow problem has $(t+1)n$ variables and n+mt constraints (where m, n and t represent the number of nodes, number of arcs and different commodities respectively).

e.g. suppose we have a problem with 100 nodes, 250 arcs and 10 commodities, then the problem will have 2750 variables and 1250 constraints.

Chapter-4

4.1 Introduction

The problem which is to determine the best way to transverse a network to get from an origin to a given destination as cheaply as possible is called Shortest Path Problem (SPP). In this problem, it is not always necessary to be concerned with measuring the length of a path, the same methods can be used with costs associated with them.

Suppose there is a distance $d\ (i, j)$ associated with the arc (i, j) in the network N. Thus, the distance from node i to node k via node j will be the sum $d\ (i, j) + d\ (j, k)$. Then the distance in the given network N, is $d(s,i_1) + d(i_1,i_2) + \ldots\ldots\ldots\ldots\ldots\ldots + d(i_p,t)$, where s is the source and t is the destination, and aim is to identify the shortest path and calculate its length.

The shortest path problem is fundamental in that it often occurs as a sub-problem of their optimization problem. The model can be applied in Transportation problem, Telephone line, Replacement problem, etc.

In the network problem discussed in chapter-2 (A Case Study of AMU Campus Network) another real problem (maintenance problem) arises. This network requires proper maintenance so that each node must function with no interruption, the solution of such a problem can be obtained by the shortest Path Technique. The problem given in Section 4.5 is a Transportation Type Network problem, it is requires a solution in the form of shortest path /route which has been obtained by the Shortest Path Algorithm.

4.2 The Mathematical Formulation of S.P.P.

Suppose, we are given a network G with m nodes and n arcs and a cost C_{ij} associated with each arc (i, j) in G. The network is such that we wish to send a single unit of flow from node 1 to node m at minimal cost. Thus, $b_1 = 1$, $b_m = -1$, and $b_i = 0$ for $I \neq 1$ or m. then, the mathematical formulation is as:

$$Minimize \sum_{i=1}^{m} \sum_{j=1}^{m} C_{ij} X_{ij}$$

$$Subjectto \sum_{j=1}^{m} X_{ij} - \sum_{k=1}^{m} X_{ki} = \begin{cases} 1 \; if & i = 1 \\ 0 \; if & i \neq 1 \; or \; m \\ -1 \; if & i = m \end{cases}$$

$$X_{ij} = 0 \; or \; 1, \qquad\qquad i, j = 1, 2, \ldots\ldots \ldots\ldots m.$$

Where the sums and the 0-1 requirements are taken over existing arcs in g.
The constraints x_{ij}=0 or 1 indicate that each arc is either in the path or not.

If we replace x_{ij} =0 or 1 by $x_{ij} \geq 0$, and if an optimal solution exists, then the simplex method would obtain an optimal integer solution.

There are several methods which have been produced for dealing with this MCF problems:

(i) Dijkstra's Algorithm
(ii) Ford's Algorithm
(iii) Floyd's Algorithm
(iv) Pollack's Algorithm.

4.3 Dijkstra's Algorithm:

The most efficient algorithm for finding the shortest directed paths from a given node (s) to all the other nodes in the network with non-negative arc lengths was first given by Dijkstra (1959).

In this method, the first step is to ensure that there is a distance associated with every pair of nodes in the network. This distance will be the arc length if there is an arc between the nodes, zero for the distance from a node to itself, and infinity for the distance between any pair of nodes which are not linked by an arc.

Dijkstra's method assigns a label to every node in the network. This label is the distance to that node from the start (s) along the shortest path found thus far. The label can be in one of the two states: it may be a permanent label, in which case the distance found is along the shortest of all paths, or it may be temporary, corresponding to some uncertainty as to whether the path found is the shortest of all. The methods gradually change temporary labels into permanent ones. The process is repeated until the terminus (t) has been assigned a permanent label, which must happen eventually, since every time the algorithm is used, one less temporary label is left, and so the number of nodes with temporary labels decreases to zero.

Algorithm

Step-0. Assign a temporary label $l(i) = \infty$ to all nodes $i \neq s$; set $l(s) = 0$ and set $p = s$, make $l(s)$ permanent. (p is the last node to be given a permanent label).

Step-1. For each node i with a temporary label, redefined $l(i)$ to be the smaller of $l(i)$ and $l(p) + d(p,i)$. Find the node i with the smallest temporary label, set p equal to this I, and make the label $l(p)$ permanent.

Step-2. If node t has a temporary label, then repeat step-1.
Otherwise, t has a permanent label, and this corresponds to the length of the shortest path from s to t through the network.
Stop.

The above Algorithms can be seen the given problem:
Find the shortest path from s to t in the given network.

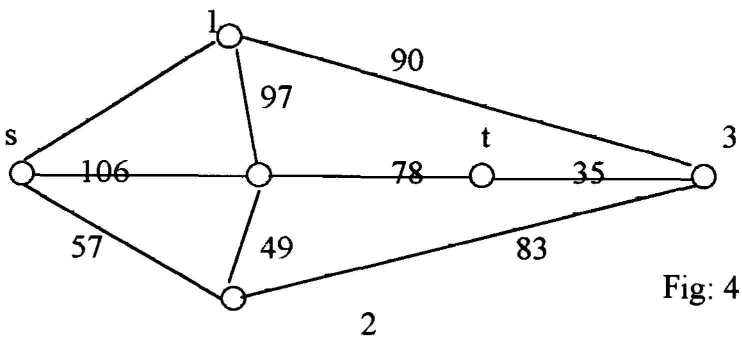


Fig: 4.1

Solution: The distance matrix from the given network is as:

Distance Matrix

$$D_{ij} : \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 29 & 57 & \infty & 106 & \infty \\ 29 & 0 & \infty & 90 & 97 & \infty \\ 57 & \infty & 0 & 83 & 49 & \infty \\ \infty & 90 & 83 & 0 & \infty & 35 \\ 106 & 97 & 49 & \infty & 0 & 78 \\ \infty & \infty & \infty & 35 & 78 & 0 \end{bmatrix} \end{array}$$

Step-0. Assign labels as in the table below:

Set p = s, make l(s) permanent,

node	s	1	2	3	4	t
l( )	0	∞	∞	∞	∞	∞
Permanent?	yes	no	no	no	no	no

Step-1. Redefined labels as below:

$l(1) = \min(\infty, 0+29) = 29^*$

$l(2) = \min(\infty, 0+57) = 57$

$l(3) = \min(\infty, 0+\infty) = \infty$

$l(4) = \min(\infty, 0+106) = 106$

$l(t) = \min(\infty, 0+\infty) = \infty$

set p=1, make l(1) permanent, because l(1) has smallest distance (cost), now

node	s	1	2	3	4	t
l( )	0	29	57	∞	106	∞
Permanent?	yes	yes	no	no	no	no

Step-2. Repeat step-1

Step-1. ……………………….

Step-2. ……………………….

……………………….

……………………….

Last step

Step-1. Redefined labels as below:

$l(t) = \min(184, 119+35) = 154$

set p = t, make l(t) permanent.

node	s	1	2	3	4	t
l( )	0	29	57	119	106	154
Permanent?	yes	yes	yes	yes	yes	yes

Step-2. Stop, The shortest path of the network from s to t is

s— 1— 2— 4— 3— t.

and shortest length is 154 units.

4.4 Generalized Shortest Paths

An important assumption in the Dijkstra's algorithm is that all the arc lengths are positive or zero. However, if the numbers associated with the arcs are costs, then it is possible for some of those to be negative. e.g. a freight haulage contractor might assign a cost to an arc on which his lorries traveled empty, and a profit (a negative cost) to arcs on which the lorries carried loads. An algorithm due to Ford allows negative arc costs to be included in the network in problems of shortest paths, provided that there are no circuits around which the net cost is negative.

4.4.1 Ford's Algorithm:

In this algorithm, labels are assigned to each of the nodes, which represent the shortest path found to the node thus far. On each iteration of the algorithm, a search is made for a label which can be reduced, using the same approach as in the Dijkstra method. The algorithm terminates when no labels can be reduced.

Algorithm

Step-0. Assign a label $l(i) = \infty$ to each node I in the network, and set $l(s) = 0$.

Step-1. For each node j, test whether there is an arc (i,j) such that $l(i) + d(i,j) < l(j)$. if there is such an arc, go to step-2. if no such arc exists for any node, stop.

Step-2. Change $l(j)$ to $l(i) + d(i,j)$.

Repeat step-1.

On termination, $l(t)$ is the length of the shortest path from s to t .

4.4.2 Floyd's Algorithm:

Since, both the algorithms (Dijkstra's and Ford's) will find the shortest paths from a given start (source) node s to all other nodes. If they were to be used n times, once with each node of the network as the start node, then the shortest path between every pair of nodes in the network would be found. Thus, Floyd's algorithm is used to find the shortest path joining any node to any other node.

In Floyd's method, nodes are numbered from 1 to n. the algorithm builds up the shortest paths between pair of nodes, first finding the shortest paths which are either direct or, which use 1 as an intermediate node, then the shortest paths which are direct or which use node 1 and/or 2 as intermediate nodes, and continue in this fashion until the shortest paths which are either direct or which use some or all of the nodes 1 to n as intermediate nodes have been found. This then provides the matrix of shortest paths. The distance found at any stage of the algorithm, say when the nodes 1 to k may be used as intermediate nodes, are sorted in an n*n matrix D_k with elements $d_k(i,j)$. ($d_k(i,j)$ is the shortest distance from I to j via some or all of the nodes 1,..........k). Then the element $d_{k+1}(i,j)$ of the next matrix will either be $d_k(i,j)$ or $d_k(i,k+1) + d_k(k+1,j)$, and the smaller of these will be chosen. The arc length of the network will form the matrix D_0 and the final matrix – the distance chart – will be D_n.

Algorithm

Step-0. Create the n*n matrix D_0 whose elements are:

$D_0(i,j) = d(i,j)$ (the length of arc (i,j), if this exist)

$= 0$ (if i=j)

$= \infty$ (if no arc (i,j) exists)

Create then n*n matrix P with elements:

$P(i,j) = i$

Set k =0

Step1-. Define the n*n matrix D_{k+1} with element $D_{k+1}(i,j) = \min(d_k(i,j), d_k(i,k+1) + d_k(k+1,j))$.

Step-2. Increase k by 1, if k = n, stop, otherwise return to step 1.

When this algorithm stops, the matrix p_0 contains integers in the range 1,……..,n; the value of p(i,j) is the last node to be visited on the shortest path from i to j, and so the path can built up (backwards) using arcs (p(i,j),j), (p(i, p(i,j)), p(i,j)),…………………… (i, p(i, p(i,……..(i,j)…….).

Consider the previous problem in Dijkstra's algorithm, we can apply this modified algorithm in the network.
Find the shortest path from s to t in the given network.

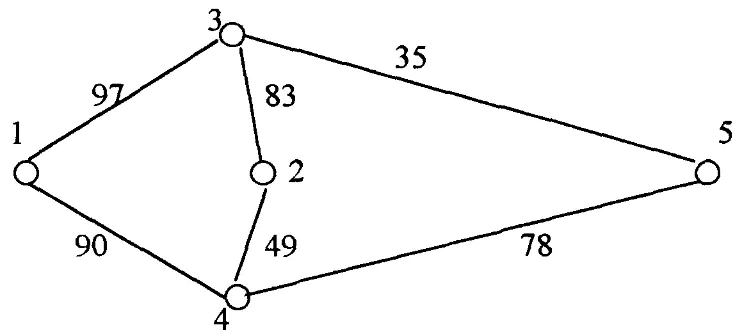


Fig:4.2

Solution: The distance matrix from the given network is as:

$$D_0 : \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & \infty & 97 & 90 & \infty \\ \infty & 0 & 83 & 49 & \infty \\ 97 & 83 & 0 & \infty & 35 \\ 90 & 49 & \infty & 0 & 78 \\ \infty & \infty & 35 & 78 & 0 \end{bmatrix} \end{array}$$

$$P : \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix} \end{array}$$

K=0,

Step-1. Using the relation

dk+1 (i,j) = min (d_k(i,j) + d_k(i,k+1) + d_k(k+1,j)).

change p(i,j) to p(k+1,j)

$d_1(1,1) = \min(0, 0+0) = 0$, $p(1,1) = 1$

$d_1(1,2) = \min(\infty, 0+0) = \infty$, $p(1,2) = 1$

$d_1(1,3) = \min(97, 0+97) = 97$, $p(1,3) = 1$

$d_1(1,4) = \min(90, 0+90) = 90$, $p(1,4) = 1$

$d_1(1,5) = \min(\infty, 0+\infty) = \infty$, $p(1,5) = 1$

$d_1(2,1) = \min(\infty, \infty+0) = \infty$, $p(2,1) = 2$,

……………………………………………

……………………………………………

……………………………………………

$d_1(5,1) = \min(\infty, \infty+0) = \infty$, $p(5,1) = 5$

$d_1(5,2) = \min(\infty, \infty+\infty) = \infty$, $p(5,2) = 5$

$d_1(5,3) = \min(35, \infty+9) = 35$, $p(5,3) = 5$

$d_1(5,4) = \min(78, \infty+90) = 78$, $p(5,4) = 5$

$d_1(5,5) = \min(0, \infty+\infty) = 0$, $p(5,5) = 5$

so,

$$D_1 = \begin{bmatrix} 0 & \infty & 97 & 90 & \infty \\ \infty & 0 & 83 & 49 & \infty \\ 97 & 83 & 0 & 187 & 35 \\ 90 & 49 & 187 & 0 & 78 \\ \infty & \infty & 35 & 78 & 0 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

Step-2. K = , (not equal to five); and do Step-1.

Step-1. Using the same method as before, we can calculate D_2 …………D_5 & corresponding P.

Last Step

Step-1. Using the same method to calculate D_5 & P to be

$$D5 = \begin{bmatrix} 0 & 139 & 97 & 90 & 132 \\ 139 & 0 & 83 & 49 & 118 \\ 97 & 83 & 0 & 113 & 35 \\ 90 & 49 & 113 & 0 & 78 \\ 132 & 118 & 35 & 78 & 0 \end{bmatrix} \quad \text{and}$$

$$P = \begin{bmatrix} 1 & 4 & 1 & 1 & 3 \\ 4 & 2 & 2 & 2 & 3 \\ 3 & 3 & 3 & 5 & 3 \\ 4 & 4 & 5 & 4 & 4 \\ 3 & 3 & 5 & 5 & 5 \end{bmatrix}$$

Step-2. K=5, stop

Now $P(2) = 3$
$P(3) = 1$
$P(4) = 2$, (or 1)
$P(5) = 3$

and so the path is (1,3) ,(3, 5)

4.5 A Case Study of S.P.P Among The Various Department in the Aligarh Muslim University (A.M.U.) Network

There are 88 departments in AMU campus that are spreaded over 467.6 hectares of land. University has very good road network which touches each departments/faculties. Among these deptts. 12 are connected with Computer Center (Chap. 2). On the basis of this network, a problem of maintenance arises, so a Maintenance Engineer will be available at Computer Center. A shortest path problem is formulated for this network through which a maintenance engineer can visit at every node so that he/she can save his/her time. Lingo software helped to approach this result obtaining the optimum route. The distance among the various departments are given in Table –1 (Chap. 2).

Discussion for the Solution of the Problem:

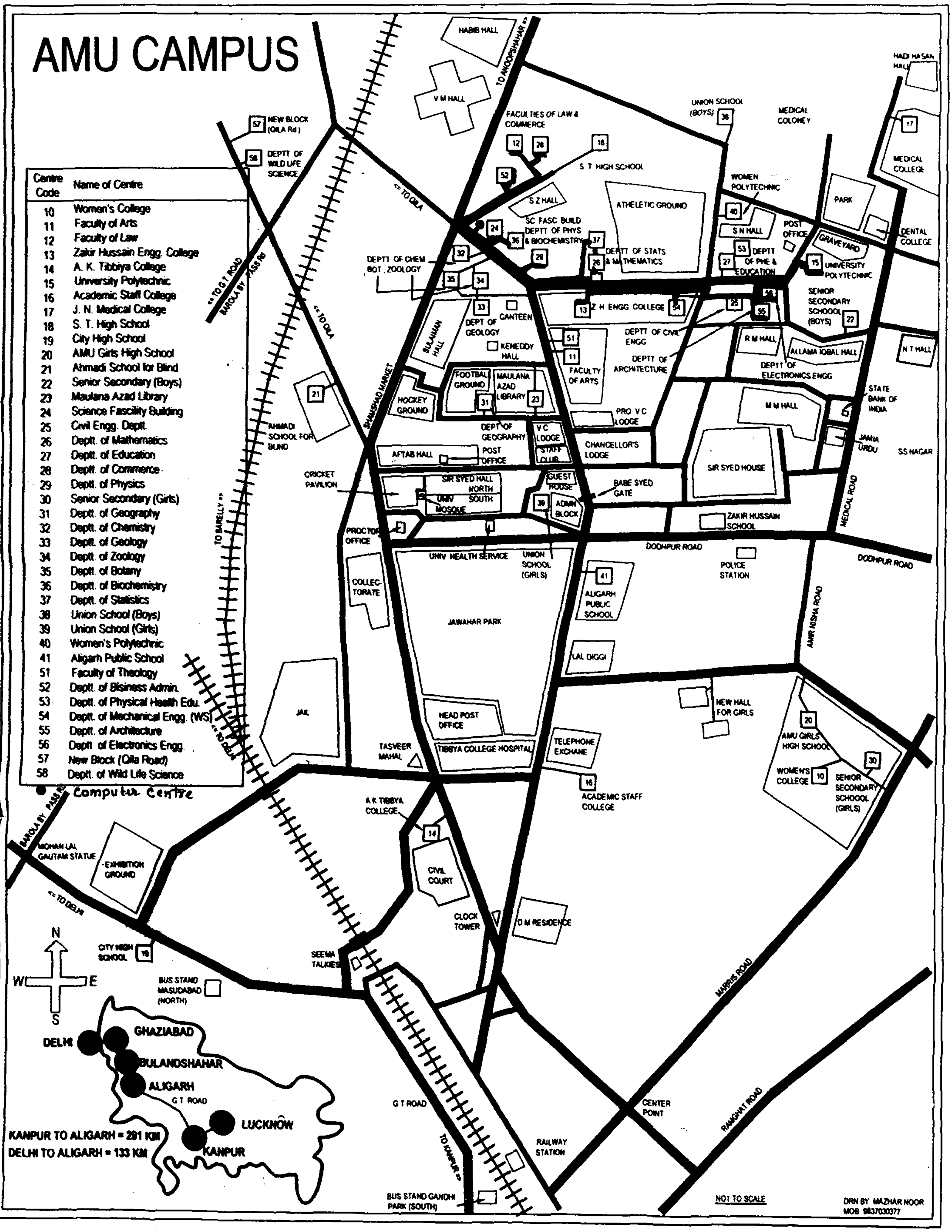
The problem under study is to find a tour that visits each node, minimizing the total distance traveled. This problem is solved by a Traveling Salesman Problem (TSP) (§ Chapter –1), we have a network of departments connected by roads. This problem lies in the fact that solution to large models tend to contain sub tours [A sub tours is a tour of a subset of nodes (cities/departments etc.) unconnected to the main tour]. One can add constraints to break the sub tour, but the number of constraints required grows dramatically as the number of nodes increase.

The above problem has been solved in two parts, part –I contained node 1 -- 11, while part – II contained node 12 -- 19.

Optimum distance of Part – I = 2647 meter
Part –II = 951.3 meter

Total optimum distance = 2647 +951.3 = 3598.3 meter = 3.598 km.

Shortest rout from Computer Center (Network Part –I) is:



Centre Code	Name of Centre
10	Women's College
11	Faculty of Arts
12	Faculty of Law
13	Zakir Hussain Engg. College
14	A. K. Tibbiya College
15	University Polytechnic
16	Academic Staff College
17	J. N. Medical College
18	S. T. High School
19	City High School
20	AMU Girls High School
21	Ahmadi School for Blind
22	Senior Secondary (Boys)
23	Maulana Azad Library
24	Science Fascility Building
25	Civil Engg. Deptt.
26	Deptt. of Mathematics
27	Deptt. of Education
28	Deptt. of Commerce
29	Deptt. of Physics
30	Senior Secondary (Girls)
31	Deptt. of Geography
32	Deptt. of Chemistry
33	Deptt. of Geology
34	Deptt. of Zoology
35	Deptt. of Botany
36	Deptt. of Biochemistry
37	Deptt. of Statistics
38	Union School (Boys)
39	Union School (Girls)
40	Women's Polytechnic
41	Aligarh Public School
51	Faculty of Theology
52	Deptt. of Business Admin.
53	Deptt. of Physical Health Edu.
54	Deptt. of Mechanical Engg. (WS)
55	Deptt. of Architecture
56	Deptt. of Electronics Engg.
57	New Block (Qila Road)
58	Deptt. of Wild Life Science
●	Computer Centre

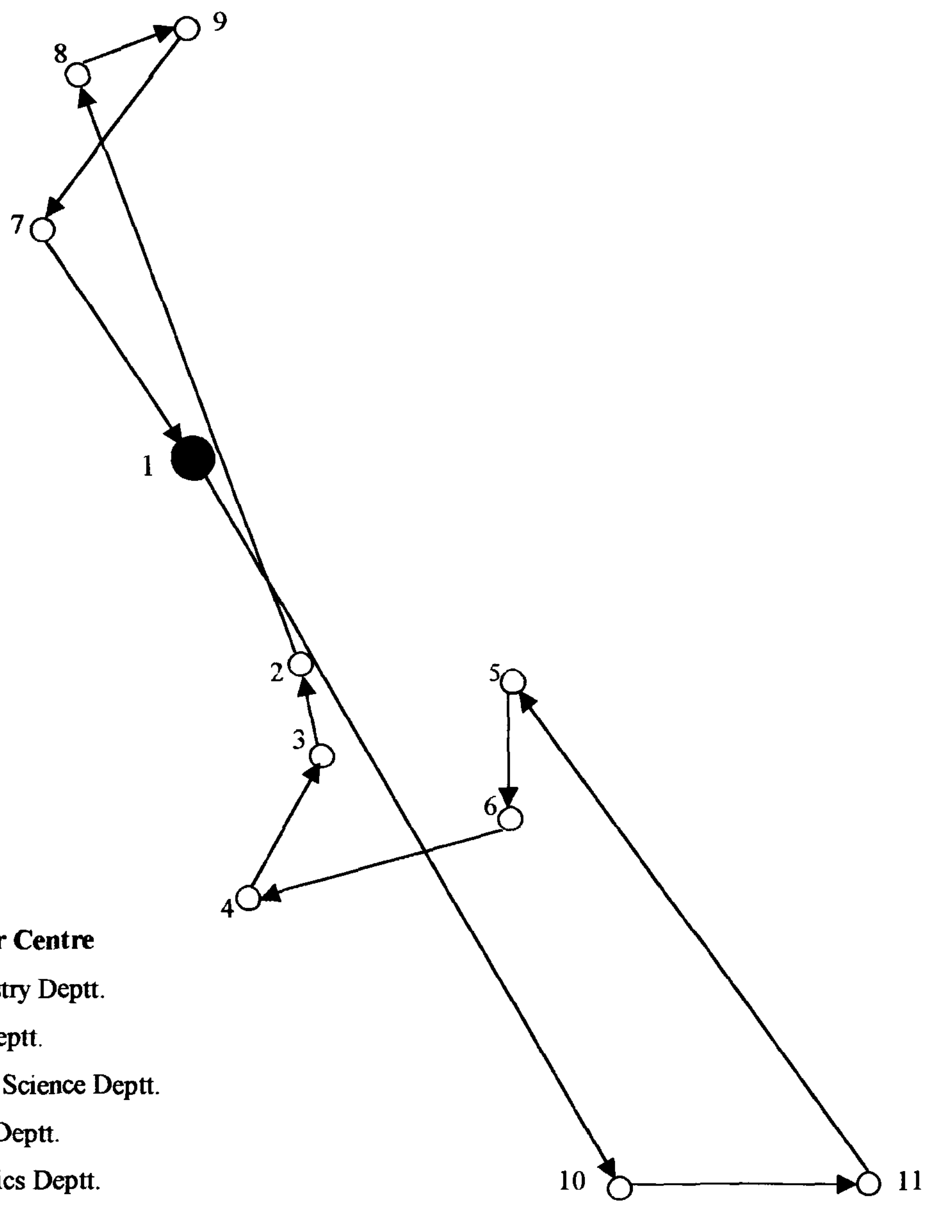
1 → 10 → 11 → 5 → 6 → 4 → 3 → 2 → 8 → 9 → 7 → 1.

Shortest route from Computer Center (Network Part –II) is:

12 → 18 → 19 → 13 → 14 → 15 → 16 → 17 → 12

The graph of optimum route for Network-I is shown in Fig. 4.3 and for Network-II is in Fig. 4.4.

Shortest Path of Network (Part -I)



1- **Computer Centre**
2- Biochemistry Deptt.
3- Physics Deptt.
4- Computer Science Deptt.
5- Statistics Deptt.
6- Mathematics Deptt.
7- Law Deptt.
8- Management and Research Centre
9- Commerce Deptt.
10- Mechanical Engg. Deptt.
11- Mechanical Engg, Workshop.

Fig. 4.3

Shortest Path of Network (Part -II)

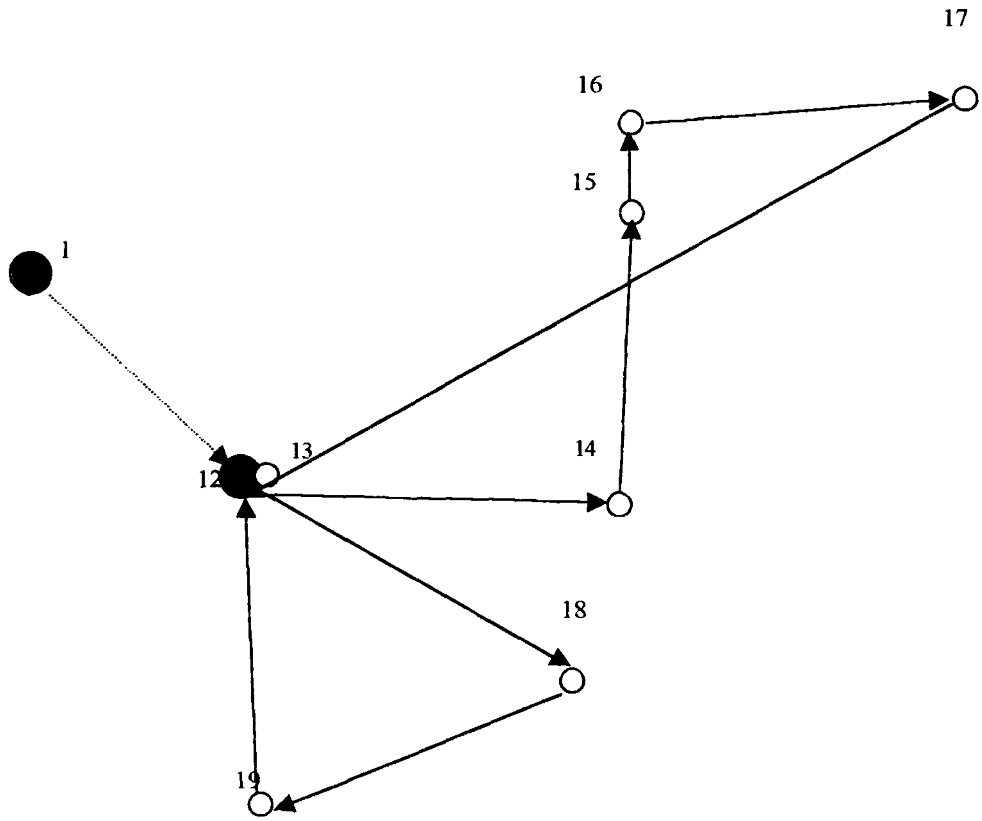


Fig. 4.4

12- Civil Engg. Deptt.
13- Applied Sciences
14- Electrical Engg. Deptt.
15- Electronic Engg. Deptt.
16- Computer Engg. Deptt.
17- University Polytechnic
18- Deptt. of Architecture
19- Petro. & Chem. Engg. Deptt.

Chapter-5

5.1 Introduction

Tours in network basically concerned with traveling at each node at once with traveling minimum distance and minimum time; where each node is connected by and return to his starting point; this problem is faced by postman in delivering mail, highway inspector, inspectors of pipelines and power cables, milk rounds men, meter readers, traveling of salesman in different cities (locations), police petrol etc.

5.2 The Postman Problem: -

The postman problem may be stated as; which route should a postman follow in order that he/she should pass along each street at least once (so as to make deliveries), and return to his starting point, while traveling the last possible total distance.

In the course of delivering mail, a postman must gradually work along the length of each of a number of inter connected roads and streets. Mail will normally be delivered at several points along each of these. Where there arc several streets meeting at a junction. The postman will have a choice of routes to follow.

The postman problem can be transformed into a network optimization problem; street can be transformed into arcs, their junctions into nodes, and the distance traveled by the postman in making deliveries along a street into the length associated with that arc, so, for a network (N.A.D.) the problem is that of identifying the circuit which traverses each arc at last once, and whose total distance is minimum. The total length of such a circuit in evaluated by summing the length of the arcs multiplied by the number of times each is used. The circuit which result will be called a Postman Tour' for the network. This problem sometimes referred to as "The Chinese Postman Problem.

5.2.1 Finding a postman tour in an undirected Network

The first stage in finding tours in undirected networks is to determine whether the network is even or not; an even network is one in which the number of arcs which are incident on every node is even. If there are any nodes with an odd number of incident arcs, then the network is described as being not even. In an even network, there will be Euler tour, and the algorithm will find one such, as the postman tour. The first stage, that of finding whether the network in even or not, is simply a matter of counting the arcs which arc incident on each node. If there is any node with an odd number, then no Euler tour exists.

Algorithm: (Postman tour in an even, undirected network),

(An Euler Tour).

Stop-0. Let s be the origin of the tour. Make all arcs 'unused'. Let t=s, it represents the last node visited. Let U and V be two empty sets of arcs, representing the partially complete tour and successive 'mini-tours' respectively.

Step-1. Find any arc between t and q (another node) which is unused. Make it used, and add it to U. Set t=q.

Step-2. If t equals s, do step 3; otherwise repeat step-1.

Step-3. Invert U in V, at the point in V where node s is first reached; U becomes empty. Find a node t which is visited in V, but has unused arcs incident on it. If there is no such node. Then stop; otherwise, set s=t, and return to step-1.

When the network is not even, then source of the arcs will have to be repeated, but the objective for the postman will be to select a set which has minimum total distance. For solving such problem algorithm is given by **Balinski.**

Algorithm:- (Postman Tour in any Undirected Network).

Determine whether the order of each node i of the network (N.A.D.) is odd or even.

Let S=(i_1. i_2,.--------------,i_{2p}) be the set of all odd - order nodes. Create the network (N.A*.D) = (N.A.D.). If S is empty go to step-3.

Step-1.

Using the matrix D of arc lengths, calculate the 2px2p matrix of shortest distances between numbers of S, using a shortest path routine.

Step-2.

Find the pairing of members of S which has minimal total length. Using this pairing, find the path which corresponds to these shortest distances, and add the arcs of this path to A*.

Step-3.

Find an Euler tour in (N.A*.D), which will be an even undirected network.'

5.2.2 Postman tours in Networks with directed arcs

In a network whose arcs are all directed, two possible situations occur: if the number of arcs entering a node is equal to the number of leaving it, for every node of the network, then there is an Euler tour; if not, there is no such a tour. The first of these situations is often referred to as being symmetric.

In a symmetric directed network, the postman tour can be found using the same method as for Euler tour is an undirected network. A tour can be built up by selecting a starting node, and leaving it by any unused arc whose sense is away from that node. The arc becomes used, and the process is repeated until the starting node is reached again. If necessary, mini-tours can be spliced into the tour, to use arcs which were unused.

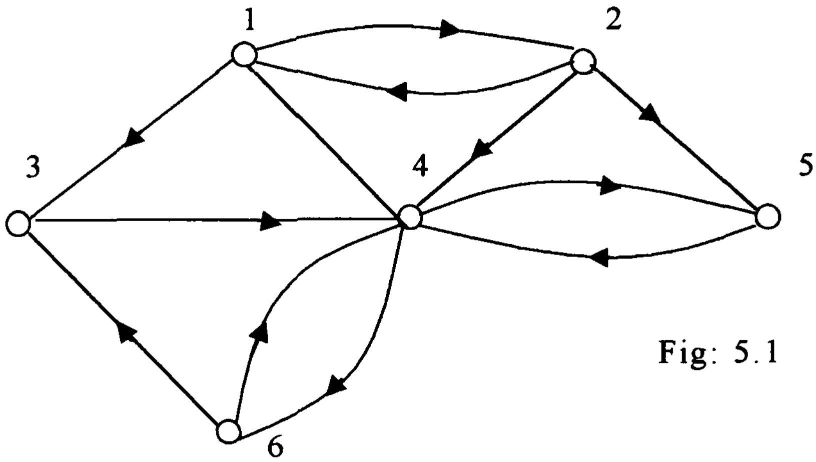


Fig: 5.1

There is no postman tour since it is impossible to return to nodes 1and 2.

Algorithm (Postman tour in any directed network).

Step-0:- For each node i in the network (N,A,D), calculate Y(i) = number of arcs leaving i-number of arcs entering i. Create the network (N,A*,D)=(N,A,D). If Y(i)=0 for all nodes i, go to step-2.

Step-1:- For each arc (i,j) in (N,A,D) create a lower bound I(i,d)=0, an upper bound U(i,j) = ∞ and a cost c(i,j)= d(i,j) (=length of the arc). Add a new node r (to be both super - source and super-sink). Add arcs as follows:

For all i with Y(i)<0, create an arc (r,i) with l(r,i)=u(r,i)= |y(i)| and c(r,i)=0 for all i with y(i) >o, create an arc (i,r) with l(,r) u(i,r)+y(i) and c (i,r) = 0. Solve the minimal cost feasible flow problem in the graph (N,A) with the given costs and constraints, resulting in flows x(i,j) in arc (i,j). Add x(i,j) copies of arc (i,j) to A*.

Step-2:- (Find an Euler tour in the directed graph (N,A*). This step is identical to the algorithm for an Euler tour in an even, undirected network except that in step-1 of that algorithm, the arc must be (t,q). It is not possible to use,. 'any arc between t and q'

5.3 The Traveling Salesman Problem

A special type of routing problem is "The Traveling Salesman Problem", in which a salesman wishes to start from a particular city visited each city once, and then return to his starting point. The objective is to select the sequence in which the cities are visited in such a way that his total traveling time is minimized. Starting from a given city, the salesman will have a total of (n-1)! different sequence (possible round trips). Further, since the salesman has to visit all the n cities, the optimal solution remains independent of selection of the starting point. In some cases the tour (which is circuit of the graph) is required to visit each town once and once only, such a circuit is known as Hamiltonian Circuit. This problem is not confined to salesman planning calls on clients; drivers of delivery vehicles who have to visit a number of customers in several localities may also wish to minimize their traveling distance, or their traveling time. The same problem also arises in cycle schedule of production or a single machine.

Formulation of the Network Problem

In much the same way as for the postman problem, the traveling salesman problems can be readily transformed into a network problem.

The towns to be visited may be identified with the node of network; the possible routes between them as the arcs of that network and the distances between towns represented as the arc lengths.

Following is the formulation of the problems whose solution will yield the minimum traveling time,

Let the variable X_{ijk} be defined as.

X_{ijk} = 1, if k^{th} directed arc is from city i to city j,

0, otherwise

Where i,j and k are integers that vary between 1 and n

Following are the constraints of the problem.

(a) Only one directed arc may be assigned to a specific k, thus,

$$\sum_{\substack{j \\ i \neq j}} \sum_{k} X_{ijk} = 1, \qquad k = 1,2,\ldots\ldots\ldots\ldots,n$$

(b) Only one other city may be reached from a specific city, thus,

$$\sum_{j} \sum_{k} X_{ijk} = 1, \qquad i = 1,2,\ldots\ldots\ldots\ldots,n$$

(c) Only one other city can initiate a direct arc to a specified city j, thus

$$\sum \sum_{k} X_{ijk} = 1, \qquad k = 1,2,\ldots\ldots\ldots\ldots,n$$

(d) Given the k^{th} directed arc ends at some specific city j, the $(k+1)^{th}$ directed arc must start at the same city j, thus

$$\sum_{i \neq j} x_{ijk} = \sum_{i \neq j} x_{jr}(k+1), \text{ for all j and k}$$

These constraints ensure that the round trip will consist of connected arcs. The objective function is to minimize.

$$Z = \sum_i \sum_j \sum_k d_{ij} x_{ijk}, \ i \neq j.$$

where d_{ij} is the distances from city i to city j.

The methods that have been developed for finding the solution to the traveling salesman problem can be divided into the two complementary groups. On one hand, there are several methods, which will always find the optimal solution, but which may require a very large number of calculations to do so, on the other hand, there are methods which can find a good solution very quickly, but which may not find the best solution at all.

5.3.1 An Approximate Method

Within the family of approximate methods, there is yet another division. Some methods will find an exact solution, given enough time, while others can find a reasonable tours, and then terminate. The method of 'two-optimality' is in the late category.

In this method, a tour is created in some arbitrary way. Then two links are broken, and the paths that are left are joined up so as to form new tour. If the length of this tour is less than the length of the original tour, then the new one is retained. Then two of its links are broken. The breaking and reassembly of links is carried out systematically, and eventually a tour is found which cannot be improved by any rearrangement is true optimal.

Algorithm

Step 0: (Create an initial tour). Selected a starting point s, choose t, so that $d(s,t) \leq d(s,j)$ for all $j \neq t$. Set $l = t$, and make nodes s and t 'visited'.

Step1: Select t from the uninvited nodes so that d(l,t) in least. Add t to the end of the tour and set l=t. If there are further un-visited nodes, repeat step1, otherwise add s to the tour and do-2.

Step 2: The visited tour will be an ordered set of nodes.

$x_1, x_2, x_3 \ldots\ldots\ldots, x_n, x_i$, with total length L. set i=1

Step 3: Set $j = i+2$.

Step 4: Consider the tour

$x_1, x_2, \ldots\ldots\ldots\ldots x_i, x_j, x_{j-1} \ldots\ldots\ldots\ldots\ldots\ldots x_{i+i}, x_{j+1}$

$x_{j+2}, \ldots\ldots\ldots\ldots x_i$

Created by exchanging links (x_i, x_{i+1}) and (x_j, x_{j+1}). If this has length less than L,

Make this the new tour, and do step-2.

Step 5: Set $j= j+1$. If $j \leq n$, do step 4. Otherwise set $i=i+1$, If $i \leq n-2$, do step 3, otherwise stop.

5.3.2 An Exact Method (Little's Algorithm)

An iterative method for the traveling salesman problem was developed by Little Murty, Sweeney and Karnel, and is exact in that it will always find the optimal salesman tour. This method is based on a branch and bound approach , which builds up a tree of nodes corresponding to particular subsets of the set of all possible tours. Associated with these nodes there are lower bounds which are less than or equal to the shortest possible salesman tour in the corresponding subset. This subset is then further divided, into two complementary subsets, and two new lower bounds found. After separate division, the subset corresponding to a node consists of just one tour, and the length of this tour is the bound associated with the node.

There are many ways that these sets and their nodes could be successfully created it, it is desirable that the number which are needed before the optimum is found be kept as small as possible. Since each node requires a certain amount of calculation to find the bound. So the iterative method 'decides' which nodes to explore (by sub-division of their sets) according to the bounds associated with the nodes.

Algorithm:

The algorithm starts with the matrix of distances between nodes, D. (This may be the matrix of shortest distances or the matrix of arc lengths). This is then transformed to give a 'reduce matrix' D′. D′ is derived from D by:

(a) setting $d(i,i) = \infty$ for all i;

(b) Subtracting from each element of D, the smallest element in the corresponding row, and when this has been done, subtracting from each resulting element the smallest element in the corresponding column. In some cases, an optimum tour can be found in D′ by inspection using zeros entry only.

In first node corresponding to all possible tours of the network, and the bound for this node is the sum of all the elements subtracted from D to obtain D′. To branch from this node, one arc(i,j) is selected, and the two nodes derived are those corresponding to all arcs which include this link as part of the tour, and those which exclude it. The arc (i, j) is one which has a zero entry in 'D' and select according to the rule 'choose that arc from the list of potential arcs whose exclusion would have the greatest effect'. This can be determined by examination of each potential arc (i,j). If this arc is not used in the tour, then the tour must use some other arc from i, and some other arc to j. Selecting the arc (i,j) with the largest such penalty cost leads to the first branch in the branch-and-bound tree. For the node which includes arc (i, j) there will be a distance matrix with the ith row and jth column removed, which can be reduced as before to give lower bound on the tours which corresponds to this set.

In this matrix, the entry for (j,i) will be set to ∞ to prevent the tour looping. Corresponding to the node for which the set is of all tours excluding the link (i,j), there will be a distance matrix for which the length of link (i,j) will be set to ∞; this too can be further reduced to give a bound on the set of all tours which correspond to this set.

This will now be two nodes from which no branches have been made. That with the smaller lower bound is selected, and the same procedure is repeated, splitting its set into two subsets, which include /exclude a particular arc, choose according to the penalty of not using it. This will create two new nodes each with a lower bound, and all the bounds on the nodes will be scanned to find the one whose bound is least. The set associated with this will then be split, and the correspond bounds calculated in the same way as for the first branch.

To summarize the successive stages of branching and evaluation of nodes in a diagram as:

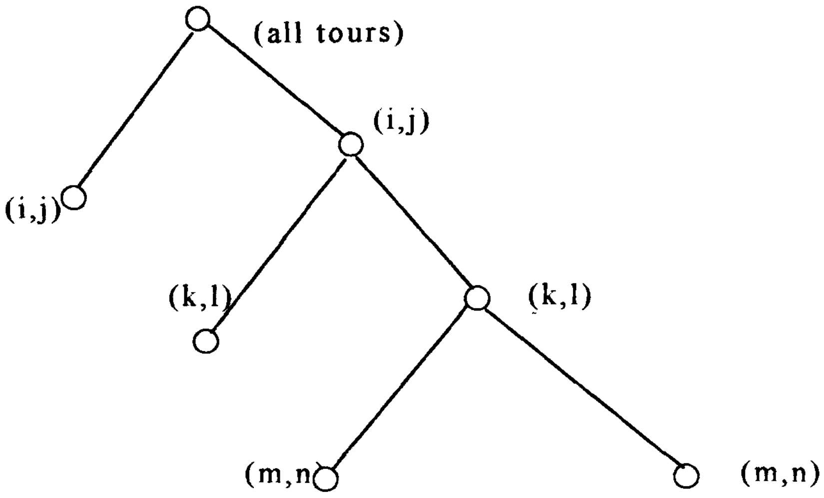


Fig: 5.2

Here, the node at the top represent the set of all tours, whose lower bound is the amount by which the original matrix D was reduced to yield D′. This gives rise to two nodes, the one on the left excluding the link (i,j) signified by $(\overline{i,j})$ and that one the right including this link. The descendants of each of these nodes include/ exclude additional links, so the node labeled $(\overline{m,n})$ in the diagram corresponds to the set of tours which include links (i,j) and (k,l) but exclude (m,n).

5.4 A Case Study of The Distribution of LPG Cylinders to the Halls/ Hostels at Aligarh Muslim University (A.M.U.) Campus

The campus of Aligarh Muslim University is spreaded in 467.6 hectares of land. It comprises of 88 departments of studies. There are 14 residential halls for students. Each Hall is constituted of more than one hostel. In a hostel there are single seated or shared rooms of varying numbers. The hall administration provides two times meal and breakfast to the students in their respective dining halls. A dining hall of a residential hall has a kitchen with good cooking facilities including big gas stoves. To supply the cooking gas (LPG Cylinder) to these 12 kitchens (2 halls for girls are not included due to being outside of the campus), there is a Central AMU gas Warehouse at some distance from the campus (see AMU Map). The distance among the Halls are given in the Table- 2.

Solution of the Problem:

This problem is formulated as Traveling Salesman Problem. After solving the problem with LINGO software, we have obtained the optimum route for distribution of LPG cylinders. The route obtained through this technique minimizes the time as well the cost of traveling. Optimum route is shown in Fig 5.4

RESULT OF TRAVELLING GAS SERVICE (A.M.U.)

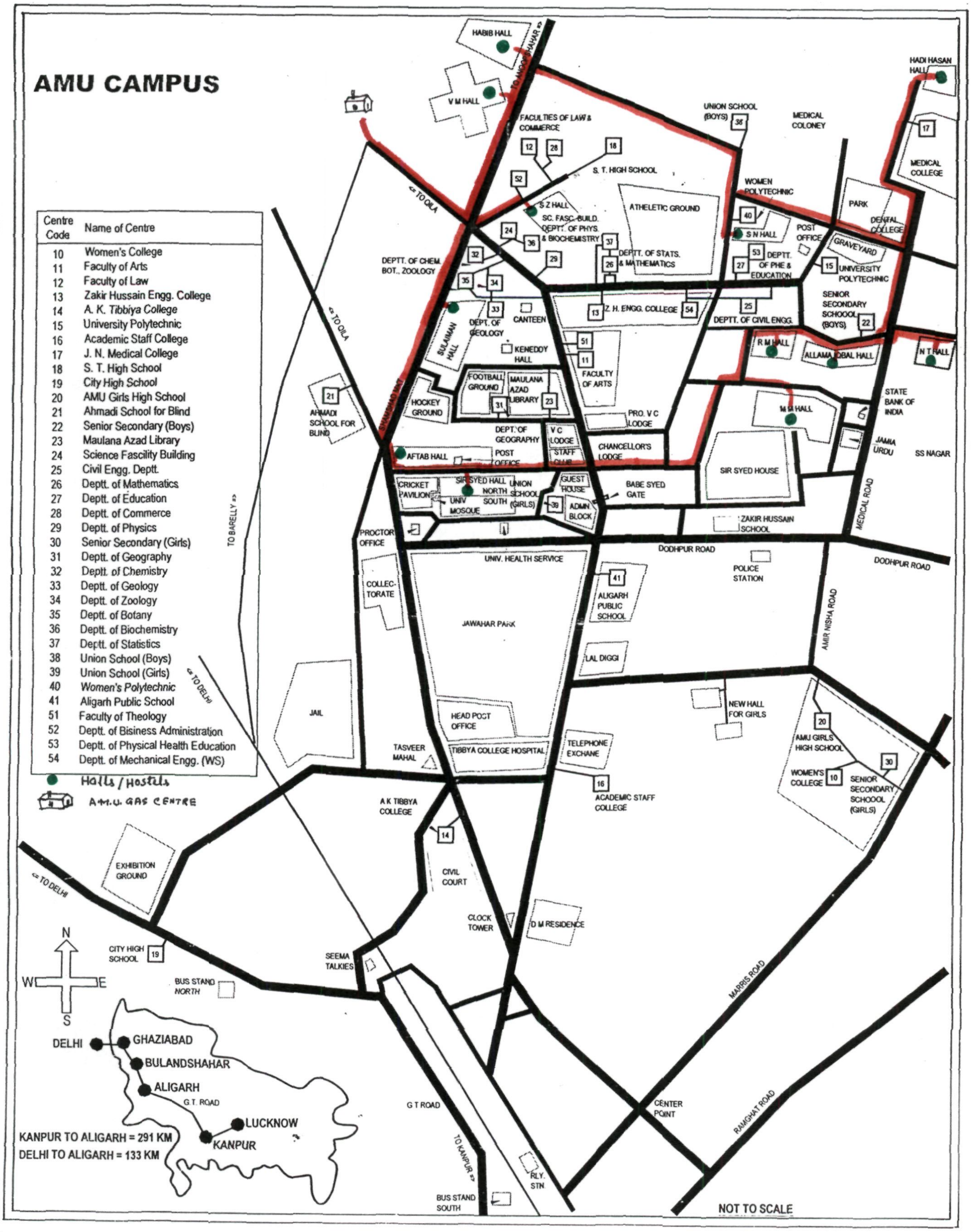
Global optimal solution found at step:	**1239**
Objective value:	**4815.000 meter**
	= **4.815 Km.**
Branch count:	**14**

Table-2

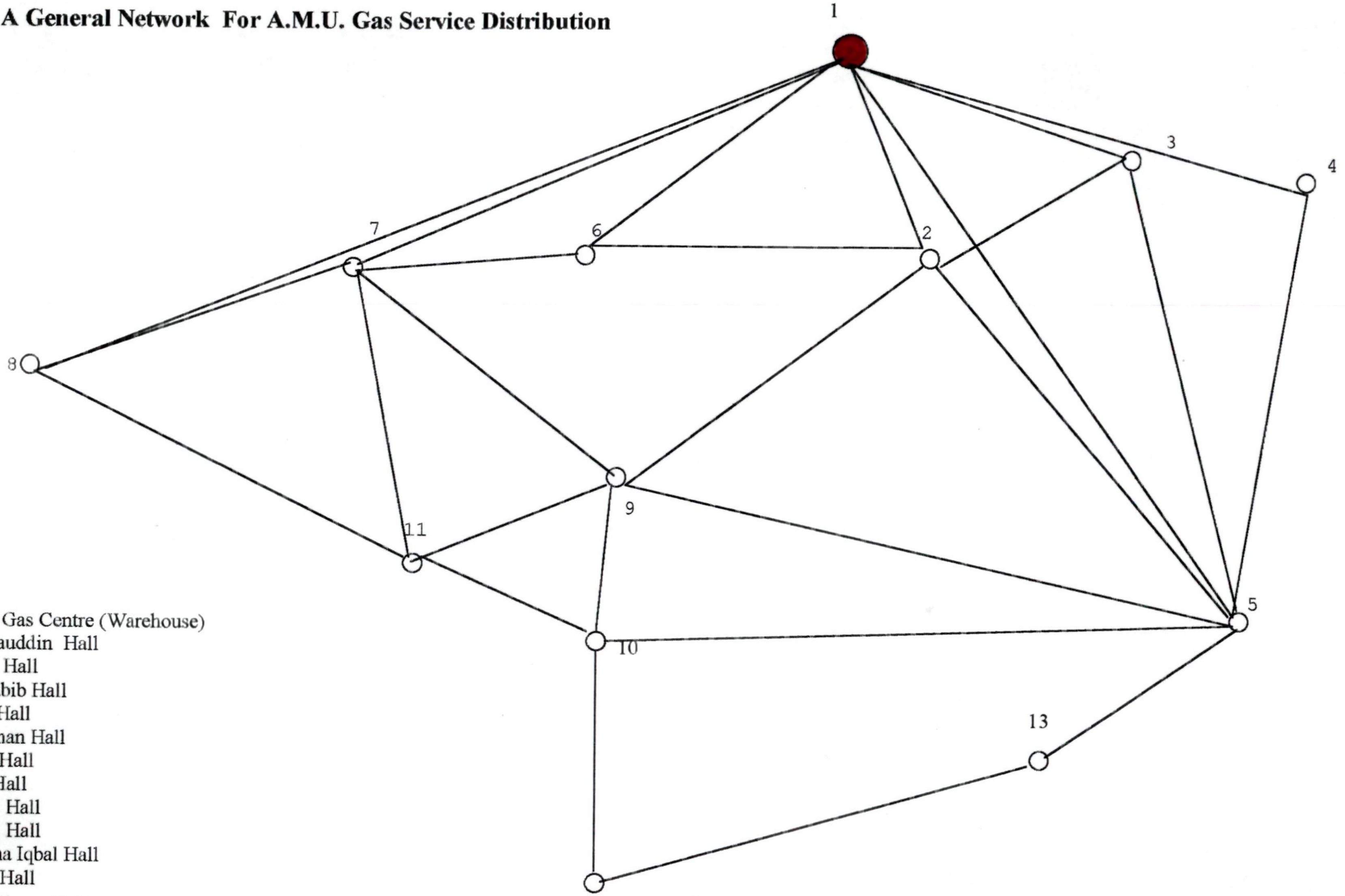
(in meters)

	1	2	3	4	5	6	7	8	9	10	11	12	13
	AMUGC	SZ	VM	MH	SN	SUL	AF	SS	RM	AI	MM	NT	HH
AMUGC	0	550	635	775	1050	665	930	1080	950	1180	1170	1470	1560
SZ	550	0	285	425	550	315	580	730	600	830	820	1120	1210
VM	635	285	0	140	490	400	665	815	685	915	905	1205	1050
MH	775	425	140	0	350	540	805	955	825	1055	1045	1345	910
SN	1050	550	490	350	0	665	930	1080	330	560	550	850	660
SUL	665	315	400	540	665	0	265	415	715	945	935	1235	1125
AF	930	580	665	805	930	265	0	150	620	850	620	1140	1280
SS	1080	730	815	955	1080	415	150	0	690	690	470	1020	1160
RM	950	600	685	825	330	715	620	690	0	230	220	520	660
AI	1180	830	915	1055	560	945	850	690	230	0	220	290	430
MM	1170	820	905	1045	550	935	620	470	220	220	0	550	650
NT	1470	1120	1205	1345	850	1235	1140	1020	520	290	550	0	540
HH	1560	1210	1050	910	660	1125	1280	1160	660	430	650	540	0

(AMUGC) AMU Gas Centre (Warehouse), (SZ) Sir Ziauddin Hall, (VM) V. M. Hall, (MH) M. Habib Hall, (SN) S. N. Hall, (SUL) Sulaiman Hall, (AF) Aftab Hall, (SS) S. S. Hall, (MM) M. M. Hall, (RM) R. M. Hall,


AMU CAMPUS
Centre Code
Name of Centre
10 Women's College
11 Faculty of Arts
12 Faculty of Law
13 Zakir Hussain Engg. College
14 A. K. Tibbiya College
15 University Polytechnic
16 Academic Staff College
17 J. N. Medical College
18 S. T. High School
19 City High School
20 AMU Girls High School
21 Ahmadi School for Blind
22 Senior Secondary (Boys)
23 Maulana Azad Library
24 Science Fascility Building
25 Civil Engg. Deptt.
26 Deptt. of Mathematics
27 Deptt. of Education
28 Deptt. of Commerce
29 Deptt. of Physics
30 Senior Secondary (Girls)
31 Deptt. of Geography
32 Deptt. of Chemistry
33 Deptt. of Geology
34 Deptt. of Zoology
35 Deptt. of Botany
36 Deptt. of Biochemistry
37 Deptt. of Statistics
38 Union School (Boys)
39 Union School (Girls)
40 Women's Polytechnic
41 Aligarh Public School
51 Faculty of Theology
52 Deptt. of Bisiness Administration
53 Deptt. of Physical Health Education
54 Deptt. of Mechanical Engg. (WS)
Halls / Hostels
A.M.U. GAS CENTRE
HABIB HALL
V M HALL
TO ANOOPSHAHAR
FACULTIES OF LAW & COMMERCE
S. T. HIGH SCHOOL
UNION SCHOOL (BOYS)
MEDICAL COLONEY
HADI HASAN HALL
MEDICAL COLLEGE
S Z HALL
SC. FASC. BUILD. DEPTT. OF PHYS & BIOCHEMISTRY
ATHELETIC GROUND
WOMEN POLYTECHNIC
S N HALL
POST OFFICE
PARK
DENTAL COLLEGE
GRAVEYARD
UNIVERSITY POLYTECHNIC
DEPTT. OF STATS. & MATHEMATICS
DEPTT. OF PHE & EDUCATION
SENIOR SECONDARY SCHOOL (BOYS)
TO QILA
DEPTT. OF CHEM. BOT., ZOOLOGY
DEPTT. OF GEOLOGY
CANTEEN
Z. H. ENGG. COLLEGE
DEPTT. OF CIVIL ENGG.
R M HALL
ALLAMA IQBAL HALL
N T HALL
SULAIMAN HALL
KENEDDY HALL
FACULTY OF ARTS
STATE BANK OF INDIA
AHMADI SCHOOL FOR BLIND
HOCKEY GROUND
FOOTBALL GROUND
MAULANA AZAD LIBRARY
PRO. V C LODGE
M M HALL
JAMIA URDU
SS NAGAR
DEPTT. OF GEOGRAPHY
V C LODGE
STAFF CLUB
CHANCELLOR'S LODGE
SIR SYED HOUSE
AFTAB HALL
POST OFFICE
CRICKET PAVILION
SIR SYED HALL NORTH SOUTH
UNIV MOSQUE
UNION SCHOOL (GIRLS)
GUEST HOUSE
ADMN. BLOCK
BABE SYED GATE
ZAKIR HUSSAIN SCHOOL
MEDICAL ROAD
TO BARELLY
PROCTOR OFFICE
DODHPUR ROAD
UNIV. HEALTH SERVICE
POLICE STATION
COLLEC-TORATE
ALIGARH PUBLIC SCHOOL
AMIR NISHA ROAD
JAWAHAR PARK
LAL DIGGI
TO DELHI
JAIL
NEW HALL FOR GIRLS
HEAD POST OFFICE
AMU GIRLS HIGH SCHOOL
TASVEER MAHAL
TIBBYA COLLEGE HOSPITAL
TELEPHONE EXCHANE
WOMEN'S COLLEGE
SENIOR SECONDARY SCHOOL (GIRLS)
A K TIBBYA COLLEGE
ACADEMIC STAFF COLLEGE
EXHIBITION GROUND
CIVIL COURT
CLOCK TOWER
D M RESIDENCE
CITY HIGH SCHOOL
SEEMA TALKIES
BUS STAND NORTH
MARRIS ROAD
N
W
E
S
DELHI
GHAZIABAD
BULANDSHAHAR
ALIGARH
G.T. ROAD
LUCKNOW
KANPUR
KANPUR TO ALIGARH = 291 KM
DELHI TO ALIGARH = 133 KM
G T ROAD
CENTER POINT
RAMGHAT ROAD
TO KANPUR
RLY. STN.
BUS STAND SOUTH
NOT TO SCALE

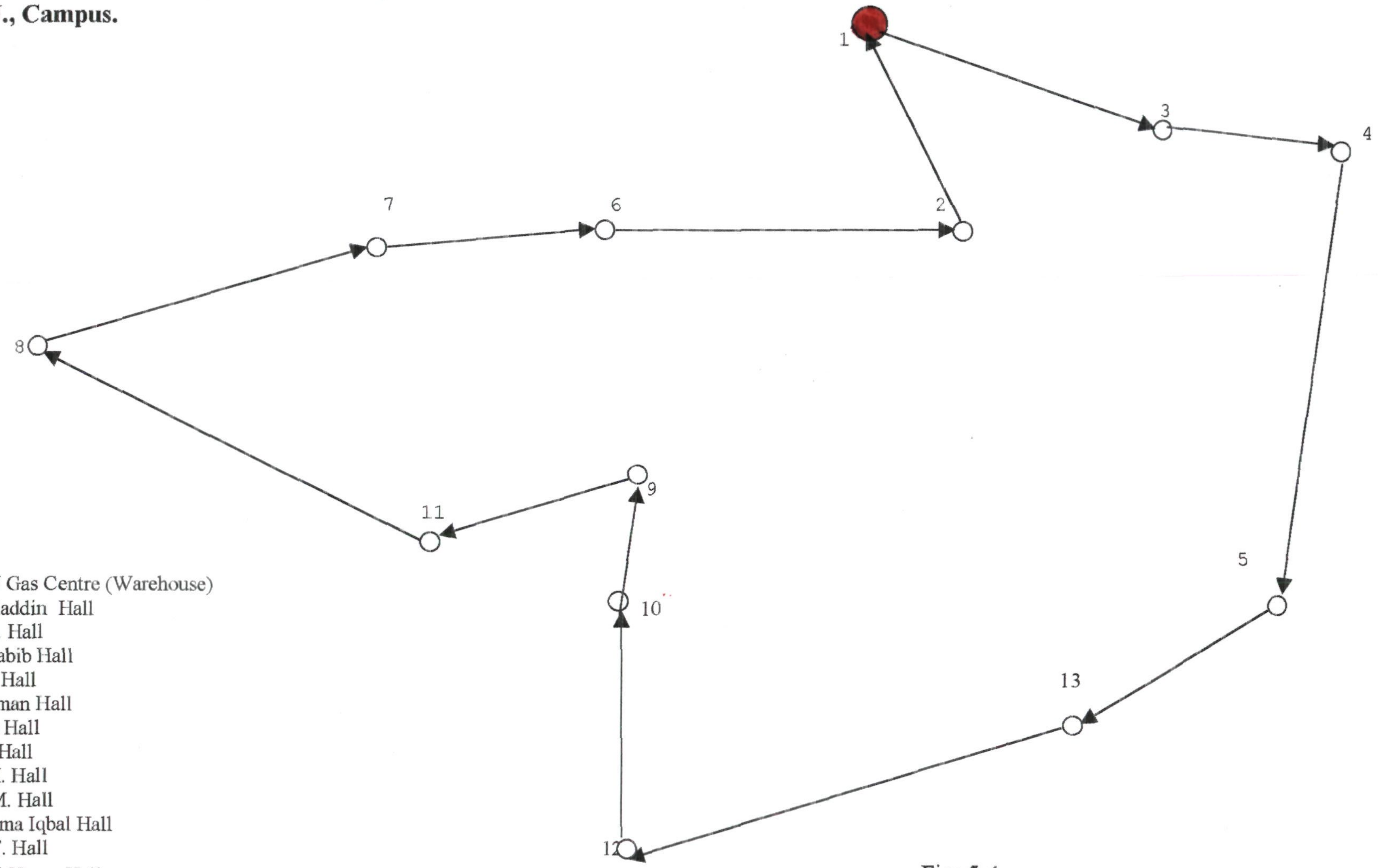
A General Network For A.M.U. Gas Service Distribution



1- AMU Gas Centre (Warehouse)
2- Sir Ziauddin Hall
3- V. M. Hall
4- M. Habib Hall
5- S. N. Hall
6- Sulaiman Hall
7- Aftab Hall
8- S. S. Hall
9- M. M. Hall
10- R. M. Hall
11- Allama Iqbal Hall
12- N. T. Hall
13- Hadi Hasan Hall.

Fig:5.3

Optimum Path for Distribution of LPG Cylinder to the Halls/Hostels at the A.M.U., Campus.



1- AMU Gas Centre (Warehouse)
2- Sir Ziaddin Hall
3- V. M. Hall
4- M. Habib Hall
5- S. N. Hall
6- Sulaiman Hall
7- Aftab Hall
8- S. S. Hall
9- M. M. Hall
10- R. M. Hall
11- Allama Iqbal Hall
12- N. T. Hall
13- Hadi Hasan Hall.

Fig: 5.4

Chapter-6

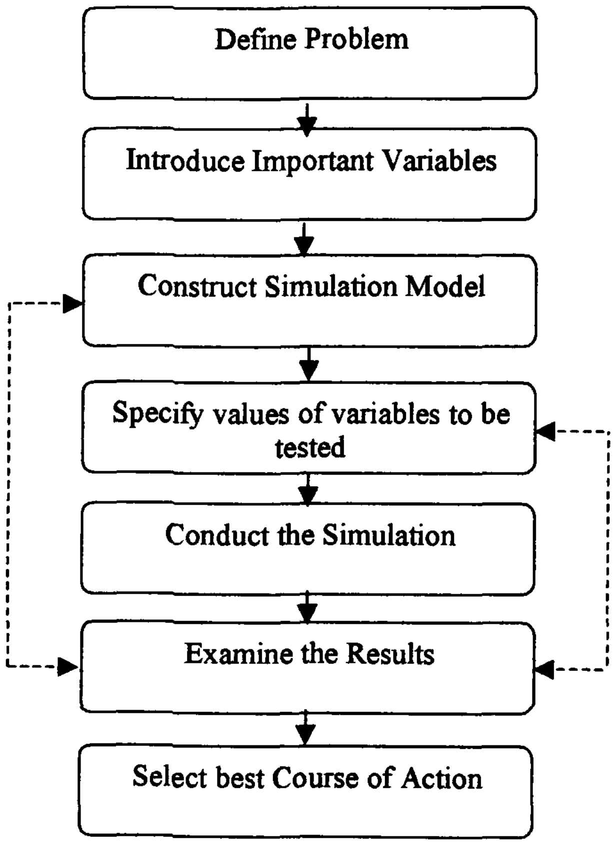
6.1 Introduction:

Simulation is a numerical technique for conducting an experiment on digital computers which involves certain type of mathematical or logical models that describe the behaviour of business or economic system (or some components) at extended period of time.

Simulation deals with both abstract and physical models.

Broadly there are four phases of the simulation process:

(a) Definition of the problem and statement of objectives
(b) Construction of an appropriate model
(c) Experimentation with the model constructed, and
(d) Evaluation of the results of Simulation.



The Methodology Of Simulation

Fig: 6.1

6.2 Why Simulation?

Using Simulation, an analyst can introduce the constants and variables related to the problem, set up the possible courses of action and establish criteria which act as measures of effectiveness. The major reasons for applying simulation technique to Operations Research Problems are as follows:

(i) Simulation is the only method available because the actual environment is difficult to observe in reality.

(ii) It may not be possible to develop an analytical solution of the problem.

(iii) Actual observation of a system is prohibitativety expensive and time consuming.

(iv) There is not sufficient time available to allow the system to operate extensively.

(v) A actual operation and observation of a system may be too disruptive.

(vi) It provides trial and error movement towards the optimal solution. The decision makers selects an alternative, experiences the effect of the selection, and then improves the selection. In this way, the selection is adjusted until it approximates the optimal solution.

(vii) Through Simulation, we can study the effect of certain informational, organizational and environmental changes on the operations of a system by making alterations to the model of the system.

(viii) Detailed observation of a system being simulated may lead to better understanding of the system and to suggest ways for improving it.

(ix) Simulation sometimes are available in the sense that they provide a convenient way of breaking down a complicated system into net-systems each at which may then be modeled by an analyst.

(x) When new components are introduced into a system, simulation can be used to help to foresee bottlenecks and other problems that may arise in the operations of the system.

Simulation Models:

A Simulation model may be a physical or mathematical model, a mental conception or a combination of them.

Broadly, the Simulation models can be classified into the following four categories:

(i) *Simulation of Deterministic Models:-*

In case of these models, the input and output variables are not permitted to be random variables and models are described by exact functional relationships.

(ii) *Simulation of Probabilistic Models: -*

These models do not take variable time into consideration.

(iii) *Simulation of Static Models: -*

These models do not take variable time into consideration.

(iv) *Simulation of Dynamic Models: -*

These models deal with time varying interaction.

6.3 Methods of Simulation:

There are two methods of Simulation:

(i) Monte Carlo Method, and
(ii) System Simulation Method.

6.3.1 Monte-Carlo Simulation:

The Monte-Carlo method is a Simulation technique in which statistical distribution function created by using a series of random numbers. This method is generally used to solve problems which cannot be adequately represented by the mathematical models, or where solution of the model is not possible by analytical method.

Monte-Carlo Simulation yields a solution which should be very close to the optimal, but not necessarily the exact solution.

The term Monte-Carlo was introduced by Van-Neuman and S.M. Ulan during World War II. As a code word for the secret work at Los Almas. It was suggested after the gambling Casino City Monte-Carlo in Monacco.

Monte-Carlo Simulation is generally computer oriented and is performed as per steps given below:

Step-1. Establish a cumulative distribution function.

Step-2. Setup the table and assign tag numbers with the help of cumulative distribution function. The tag numbers are assigned in such a way as to reflect the probability of the various events.

Step-3. Obtain the random numbers from a random number table. When using a random number table one can start at any point in the table and proceed in any direction, but one should not choose or select the random numbers.

Uses: Monte-Carlo method was used to evaluate complex multi-dimensional integrals and to solve certain integral equations occurring in Physics that did not have analytical solution. Monte-Carlo method can be used not only for solution of Stochastic problems but also for solution of deterministic problems.,

6.3.2 System Simulation Method:

System Simulation Method is one wherein there is reproduction of the operating environment and the system allows for analyzing the response from the environment to alternative management actions. This method is quite complicated and well tend to be costly.

6.4 Solution of the Problem of Distribution of LPG Cylinders Through Simulation:

The problem discussed in Chapter-5 is solved with the new method (Simulation technique). There are 3 means of transport in AMU Gas Service Centre (i.e. Tata 407, Bicycle and 3 Wheeler). They are delivering gas cylinders to different halls in varying duration of time shown in fig 5.3. Table-3 gives the distribution route under the column of activity. Next column in the table provides the respective duration of time taken by the three means of transport for each route. For the three durations of time corresponding to one activity the probability is computed by dividing the individual time duration by the total of the three timings (upto two significant decimal places). These probabilities are given under the column probability of Table-3. Expected time $\sum_{i=1}^{3} p_i x_i$ duration is computed for each activity. Cumulative probabilities for each activity are also shown in the Table. Tag numbers are computed as follows with the assumption of first tag no. as 00. End of the Tag No. = 100 * Cumulative prob. - 1.

Now, for this problem, we simulated the duration of delivering in 10 times (using the random numbers) and estimated the average duration of delivering the cylinders to various halls. The activities, 1 to 13, shown in Fig. 5.3.

Table –3

Activity	Time (in Minute) x_i	Probability p_i	Expected Duration $\sum_{i=1}^{3} p_i x_i$	Cumulative Probability Distribution	Tag Numbers
1 -3	28	0.28		0.28	00-27
	40	0.41	33.54	0.69	28-68
	30	0.31		1.00	69-99
3 -4	10	0.27		0.27	00-26
	15	0.41	12.69	0.68	27-67
	12	0.32		1.00	68-99
4 -5	20	0.27		0.27	00-26
	30	0.40	25.65	0.67	27-66
	25	0.33		1.00	67-99
5 -13	30	0.29		0.29	00-28
	42	0.40	44.12	0.69	29-68
	32	0.31		1.00	69-99
13 -12	25	0.28		.28	00-27
	35	0.39	37.55	.67	00-27
	30	0.33		1.0	67-99
12 – 10	12	0.27		0.27	00-26
	18	0.40	15.39	0.67	27-66
	15	0.33		1.00	67-99
10 – 9	10	0.26		0.26	00-25
	16	0.41	13.45	0.67	26-66
	13	0.33		1.00	67-99
9 -11	10	0.26		0.26	00-25
	15	0.39	12.87	0.69	26-64
	13	0.34		1.00	65-99
11 – 8	22	0.26		0.26	00-25
	33	0.40	28.44	0.66	26-65
	28	0.34		1.00	66-99
8 – 7	11	0.28		0.28	00-27
	16	0.40	13.64	0.68	28-67
	13	0.32		1.00	68-99
7 – 6	11	0.26		0.26	00-25
	17	0.41	14.45	0.67	26-66
	14	0.33		1.00	67-99

6 – 2	18	0.26		0.26	00-25
	28	0.41	23.75	0.67	26-66
	23	0.33		1.00	67-99
2 – 1	26	0.28		0.28	00-27
	36	0.39	31.55	0.67	28-66
	31	0.33		1.00	67-99

The simulated time for delivering the LPG cylinders through the network, Fig 5.4, with the time duration given in Table- 3 are computed – see Table- 4. The first column of Table-4 has the serial number of ten iterations of simulations. Next thirteen columns contain the Tag No. (RD) picked up from Random Number Table and their corresponding time duration (D) for each activity. For instance for activity 1-3 the random no. is 17 which falls in the range of 00-27 of Tag No. of Table-3 with corresponding time duration 28 minutes, average duration time, in minutes is computed for all the thirteen columns under 'D'. The last row of Table-4 is copied from the fourth column of Table-3 to compute the average duration of time obtained by the simulation and the expected duration time.

The total duration of time for delivering LPG Cylinders is 4.8 hours by the simulation technique while it is 4.87 hours by the theoretical procedure. The precision of estimate obtained by the simulation technique concludes the utility of simulation in traveling salesman problem

Iterations

Table-4

Simulatic Run	1 - 3 RD	D	3 – 4 RD	D	4 - 5 RD	D	5 – 13 RD	D	13 – 12 RD	D	12 – 10 RD	D	10 – 9 RD	D	9 - 11 RD	D	11 – 8 RD	D	8 – 7 RD	D	7 – 6 RD	D	6 – 2 RD	D	2 – 1 RD	D
1	17	28	77	12	14	20	26	30	11	25	26	12	67	13	73	13	64	33	45	16	87	14	01	18	36	36
2	41	40	71	12	77	25	52	42	87	30	72	15	25	10	47	15	15	22	50	16	27	17	34	28	24	26
3	89	30	39	15	11	20	47	42	24	25	38	18	14	10	99	13	24	22	85	13	27	17	56	28	03	26
4	00	28	74	12	75	25	88	32	18	25	72	15	92	13	87	13	81	28	15	11	12	11	73	23	98	31
5	79	30	80	12	91	25	76	32	79	30	44	18	99	13	87	13	38	33	93	13	22	11	52	28	72	31
6	84	30	07	10	65	30	71	32	47	35	50	18	14	10	14	10	58	33	40	16	74	14	87	23	37	36
7	59	40	42	15	41	30	79	32	51	35	49	18	96	13	70	13	27	33	76	13	45	17	36	28	46	36
8	32	40	40	15	59	30	16	30	45	35	33	18	77	13	15	10	43	33	44	16	20	11	31	28	05	26
9	02	28	04	10	23	20	42	42	29	35	86	15	92	13	62	15	93	28	11	11	60	17	56	28	41	36
10	88	30	43	15	86	25	79	32	38	35	59	18	82	13	68	13	28	33	92	13	95	14	78	23	84	31
Average duration (in minutes)		32.4		12.8		25.0		34.6		31.0		16.5		12.1		12.8		29.1		13.8		14.3		25.5		31.5
Expected duration (in minutes)		33.54		12.77		25.65		35.42		30.55		15.39		13.45		12.87		28.44		13.64		14.45		23.75		31.50

Total Duration of Delivery Required: 291.82 minutes = 4.8 hours

Total Expected Duration : = $\sum p_i x_i$, where p_i = probability and x_i = duration.

= 292.42 minutes = 4.87 hours

Conclusion

Conclusion

Many types of network optimization problems have been elaborated in this dissertation. The algorithms to solve these problems have also been described in detail. Clear comprehension of these techniques required that certain real life situations be tackled with by these. Hence a Minimal Spanning Tree is developed for installing an optical fibre based campus network as a part of Phase-I of Campus Wide Local Area Network linking 23 Departments to the Computer Centre of Aligarh Muslim University (A.M.U.). The minimal spanning tree thus obtained provided optimum usage of optical fibre.

The Shortest Path Problem along with the algorithms to solve it is described in the dissertation. For implementing these techniques a situation is discussed in which the shortest path for the maintenance engineer visiting the above mentioned network is obtained.

Problem of tours in network has also been explained in the dissertation. Mainly the Postman problem, Traveling Salesman problem and their respective algorithms have been discussed. The distribution of L.P.G. cylinders from the A.M.U. Gas Agency to various halls of residences of students of A.M.U. is considered for obtaining an optimum tour in network.

Simulation techniques being an important tool for research are described with their implementation in the above real life situation of distribution of L.P.G. Cylinders.

Additionally the Minimum Cost Flow Problem and the relevant methods are also covered in the dissertation.

LINGO, a software for optimization methods, is used for the above mentioned data analyses.

By studying the network optimization problems and their implementation for some real life situation along with the experience of simulation techniques and that of LINGO it is found that enormous scope for further research is available in this area.

Computer Programes in LINGO

APPENDIX – A:

COMPUTER PROGRAMMES FOR THE CASE STUDIES IN LINGO

A Case Study Of Optical Fibre Based Campus Wide Network Phase -I of Aligarh Muslim University (A.M.U.).

PART-1

```
SETS:
        CITY: LVL;
        !LVL(I)=LEVEL OF CITY I IN TREE, LVL(1)=0;
        LINK(CITY,CITY):
        DIST, !THE DISTANCE MATRIX;
        X; !X(I,J)=1 IF WE USE LINK I,J;
ENDSETS
DATA:
CITY = CC   BIOCH  PHY   CS  STATS MATHSLAW  MBA   COMM   MED  MEDW;  !CITY 1 IS BASED;
DIST = 0     46    207   206 442   391   435   390   490    412  528  !FROM CC;
       46     0     50   251 571   521   481   435   536    541  671  !FROM BIOCH;
      208    50      0    50 370   319   642   597   997    350  456  !FROM PHY;
      206   251     50     0 320   269   641   596   696    290  406  !FROM CS;
      442   571    370   320   0   30    877   832   932    610  726  !FROM STATS;
      391   521    319   269 30    0     826   781   881    559  675  !FROM MATHS;
      435   481    642   641 961   826   0     110   30     847  963  !FROM LAW;
      390   436    597   596 916   781   110   0     80     802  918  !FROM MBA;
      490   536    697   696 932   881   30    80    0      902  1018 !FROM COMM;
      412   541    340   290 610   559   847   802   902    0    130  !FROM MED;
      528   671    456   406 726   675   963   918   1018   130  0;   !FROM MEDW;
ENDDATA
! THE MODEL SIZE N>=8;
        N=@SIZE(CITY);
        MIN=@SUM(LINK:DIST*X);
        @FOR(CITY(K)| K#GT#1:
        @SUM(CITY(I)| I#NE#K: X(I,K))=1;
        @FOR (CITY(J)| J#NE#K:
        LVL(K)>=LVL(J)+X(J,K)
                -(N-2)*(1-X(J,K))
```

```
              +(N-3)*X(K,J););ёk);
```

PART-2

```
SETS:
        CITY: LVL;
        !LVL(I)=LEVEL OF CITY I IN TREE, LVL(1)=0;
        LINK(CITY,CITY):
        DIST, !THE DISTANCE MATRIX;
        X; !X(I,J)=1 IF WE USE LINK I,J;
ENDSETS
DATA:
CITY =  CED    AS     ELCAL  ELNIC  COMP   UPOLY  BARCH  CHE_PAT;  !CITY 1 IS BASED;
DIST =  0.0    0.0    76.2   131.1  137.2  297.2  137.2  164.6     !FROM CED;
        0.0    0.0    76.2   131.1  137.2  297.2  137.2  164.6     !FROM AS;
        76.2   76.2   0.0    50.0   56.1   216.1  213.4  240.8     !FROM ELCAL;
        131.1  131.1  50.0   0.0    6.1    166.1  268.3  295.7     !FROM ELNIC;
        137.2  137.2  56.1   6.1    0.0    160.0  274.4  301.8     !FROM COMP;
        297.2  297.2  216.1  166.1  160.0  0.0    434.4  461.8     !FROM UPOLY;
        137.2  137.2  213.4  268.3  274.4  434.4  0.0    60.0      !FROM BARCH;
        164.6  164.6  240.8  295.7  301.8  461.8  60.0   0.0;      !FROM CHE_PAT;
        ENDDATA
! THE MODEL SIZE N>=8;
        N=@SIZE(CITY);
        MIN=@SUM(LINK:DIST*X);
        @FOR(CITY(K)| K#GT#1:
        @SUM(CITY(I)| I#NE#K: X(I,K))=1;
        @FOR (CITY(J)| J#NE#K:
        LVL(K)>=LVL(J)+X(J,K)
              -(N-2)*(1-X(J,K))
              +(N-3)*X(K,J);););
LVL(1)=0;       !CITY 1 HAS LEVEL 0;
        @SUM(CITY(J)| J#GT#1:X(1,J))>=1;
```

```
@FOR(LINK:@BIN(X););
@FOR(CITY(K)| K#GT#1:
@BND(1, LVL(K), 9999999);
LVL(K)<=N-1-(N-2)*X(1,K););
END
```

A Case Study of S.P.P. among the various Department in the Aligarh Muslim University (A.M.U.) Aligarh.

!Shortest Path Problem for the COMPUTER CNETRE PART - I;

```
SETS:
 CITY / 1..11/: U; ! U( I) = sequence no. of city;
 LINK( CITY, CITY):
      DIST,  ! The distance matrix;
         X;  ! X( I, J) = 1 if we use link I, J;
ENDSETS
DATA:   !Distance matrix, it need not be symmetric;
!CITY =CC   BIOCH PHY  CS   STATS MATHS LAW  MBA  COMM MED  MEDW;
!CITY 1 IS BASED;
DIST =  0    46   207  206  442  391  435  390  490  412  528  !FROM CC;
       46     0    50  251  571  521  481  435  536  541  671  !FROM BIOCH;
      208    50     0   50  370  319  642  597  997  350  456  !FROM PHY;
      206   251    50    0  320  269  641  596  696  290  406  !FROM CS;
      442   571   370  320    0   30  877  832  932  610  726  !FROM STATS;
      391   521   319  269   30    0  826  781  881  559  675  !FROM MATHS;
      435   481   642  641  961  826    0  110   30  847  963  !FROM LAW;
      390   436   597  596  916  781  110    0   80  802  918  !FROM MBA;
      490   536   697  696  932  881   30   80    0  902 1018  !FROM COMM;
      412   541   340  290  610  559  847  802  902    0  130  !FROM MED;
      528   671   456  406  726  675  963  918 1018  130    0; !FROM MEDW;

ENDDATA
!The model:Ref. Desrochers & Laporte, OR Letters,  Feb. 91;
 N = @SIZE( CITY);
 MIN = @SUM( LINK: DIST * X);
 @FOR( CITY( K):
 !  It must be entered;
  @SUM( CITY( I)| I #NE# K: X( I, K)) = 1;
```

```
! It must be departed;
 @SUM( CITY( J)| J #NE# K: X( K, J)) = 1;
! Weak form of the subtour breaking constraints;
! These are not very powerful for large problems;
 @FOR( CITY( J)| J #GT# 1 #AND# J #NE# K:
     U( J) >= U( K) + X ( K, J) -
     ( N - 2) * ( 1 - X( K, J)) +
     ( N - 3) * X( J, K));
);
! Make the X's 0/1;
@FOR( LINK: @BIN( X));
! For the first and last stop we know...;
@FOR( CITY( K)| K #GT# 1:
 U( K) <= N - 1 - ( N - 2) * X( 1, K);
 U( K) >= 1  + ( N - 2) * X( K, 1)
);
END
```

```
!Shortest Path Problem for the COMPUTER CNETRE PART - II;

SETS:
 CITY / 1..8/: U; ! U( I) = sequence no. of city;
 LINK( CITY, CITY):
      DIST,  ! The distance matrix;
         X;  ! X( I, J) = 1 if we use link I, J;
ENDSETS
DATA:
 !Distance matrix, it need not be symmetric;
!CITY =CED   AS    ELCAL  ELNIC  COMP   UPOLY   BARCH  CHE_PAT;
!CITY 1 IS BASED;
DIST = 0.0    0.0    76.2   131.1  137.2  297.2   137.2  164.6   !FROM CED;
       0.0    0.0    76.2   131.1  137.2  297.2   137.2  164.6   !FROM AS;
       76.2   76.2   0.0    50.0   56.1   216.1   213.4  240.8   !FROM ELCAL;
       131.1  131.1  50.0   0.0    6.1    166.1   268.3  295.7   !FROM ELNIC;
       137.2  137.2  56.1   6.1    0.0    160.0   274.4  301.8   !FROM COMP;
       297.2  297.2  216.1  166.1  160.0  0.0     434.4  461.8   !FROM UPOLY;
       137.2  137.2  213.4  268.3  274.4  434.4   0.0    60.0    !FROM BARCH;
       164.6  164.6  240.8  295.7  301.8  461.8   60.0   0.0;    !FROM CHE_PAT;

ENDDATA
!The model:Ref. Desrochers & Laporte, OR Letters,  Feb. 91;
 N = @SIZE( CITY);
 MIN = @SUM( LINK: DIST * X);
```

```
@FOR( CITY( K):
!  It must be entered;
 @SUM( CITY( I)| I #NE# K: X( I, K)) = 1;
!  It must be departed;
 @SUM( CITY( J)| J #NE# K: X( K, J)) = 1;
! Weak form of the subtour breaking constraints;
! These are not very powerful for large problems;
 @FOR( CITY( J)| J #GT# 1 #AND# J #NE# K:
     U( J) >= U( K) + X ( K, J) -
     ( N - 2) * ( 1 - X( K, J)) +
     ( N - 3) * X( J, K));
);
! Make the X's 0/1;
@FOR( LINK: @BIN( X));
! For the first and last stop we know...;
@FOR( CITY( K)| K #GT# 1:
 U( K) <= N - 1 - ( N - 2) * X( 1, K);
 U( K) >= 1  + ( N - 2) * X( K, 1)
);
END
```

A Case Study of the Distribution of LPG Cylinders to the Halls/Hostels at the Aligarh Muslim University (A.M.U.) Campus

```
SETS:
 CITY / 1..13/: U; ! U( I) = sequence no. of city;
 LINK( CITY, CITY):
    DIST,  ! The distance matrix;
      X;  ! X( I, J) = 1 if we use link I, J;
ENDSETS
DATA:  !Distance matrix, it need not be symmetric;
!HALL=AMUGC  SZ   VM   MH   SN    SUL  AF   SS    RM   AI    MM   NT    HH;
DIST= 0      550  635  775  1050  665  930  1080  950  1180  170  1470  1560
      550    0    285  425  550   315  580  730   600  830   820  1120  1210
      635    285  0    140  490   400  665  815   685  915   905  1205  1050
```

```
775     425    140    0      350    540    805    955    825    1055   1045   1345   910
1050    550    490    350    0      665    930    1080   330    560    550    850    660
665     315    400    540    665    0      265    415    715    945    935    1235   1125
930     580    665    805    930    265    0      150    620    850    620    1140   1280
1080    730    815    955    1080   415    150    0      690    690    470    1020   1160
950     600    685    825    330    715    620    690    0      230    220    520    660
1180    830    915    1055   560    945    850    690    230    0      220    290    430
1170    820    905    1045   550    935    620    470    220    220    0      550    650
1470    1120   1205   1345   850    1235   1140   1020   520    290    550    0      540
1560    1210   1050   910    660    1125   1280   1160   660    430    650    540    0;
ENDDATA
!The model:Ref. Desrochers & Laporte, OR Letters, Feb. 91;
 N = @SIZE( CITY);
 MIN = @SUM( LINK: DIST * X);
 @FOR( CITY( K):
 ! It must be entered;
  @SUM( CITY( I)| I #NE# K: X( I, K)) = 1;
 ! It must be departed;
  @SUM( CITY( J)| J #NE# K: X( K, J)) = 1;
 ! Weak form of the subtour breaking constraints;
 ! These are not very powerful for large problems;
  @FOR( CITY( J)| J #GT# 1 #AND# J #NE# K:
     U( J) >= U( K) + X ( K, J) -
     ( N - 2) * ( 1 - X( K, J)) +
     ( N - 3) * X( J, K));
 );
 ! Make the X's 0/1;
 @FOR( LINK: @BIN( X));
 ! For the first and last stop we know...;
 @FOR( CITY( K)| K #GT# 1:
  U( K) <= N - 1 - ( N - 2) * X( 1, K);
  U( K) >= 1  + ( N - 2) * X( K, 1)
 );
END
```

Iterative Computations

APPENDIX – B:

ITERATIVE COMPUTATIONS FOR THE PROBLEMS

(Minimal Spanning Tree) Network -I

Global optimal solution found at step: 288
Objective value: 1365.000
Branch count: 17

Variable	Value	Reduced Cost
N	11.00000	0.0000000
LVL(CC)	0.0000000	0.0000000
LVL(BIOCH)	1.000000	0.0000000
LVL(PHY)	2.000000	0.0000000
LVL(CS)	3.000000	0.0000000
LVL(STATS)	5.000000	0.0000000
LVL(MATHS)	4.000000	0.0000000
LVL(LAW)	3.000000	0.0000000
LVL(MBA)	1.000000	0.0000000
LVL(COMM)	2.000000	0.0000000
LVL(MED)	4.000000	0.0000000
LVL(MEDW)	5.000000	0.0000000
DIST(CC, CC)	0.0000000	0.0000000
DIST(CC, BIOCH)	46.00000	0.0000000
DIST(CC, PHY)	207.0000	0.0000000
DIST(CC, CS)	206.0000	0.0000000
DIST(CC, STATS)	442.0000	0.0000000
DIST(CC, MATHS)	391.0000	0.0000000
DIST(CC, LAW)	435.0000	0.0000000
DIST(CC, MBA)	390.0000	0.0000000
DIST(CC, COMM)	490.0000	0.0000000
DIST(CC, MED)	412.0000	0.0000000
DIST(CC, MEDW)	528.0000	0.0000000
DIST(BIOCH, CC)	46.00000	0.0000000
DIST(BIOCH, BIOCH)	0.0000000	0.0000000
DIST(BIOCH, PHY)	50.00000	0.0000000
DIST(BIOCH, CS)	251.0000	0.0000000
DIST(BIOCH, STATS)	571.0000	0.0000000
DIST(BIOCH, MATHS)	521.0000	0.0000000
DIST(BIOCH, LAW)	481.0000	0.0000000
DIST(BIOCH, MBA)	435.0000	0.0000000
DIST(BIOCH, COMM)	536.0000	0.0000000
DIST(BIOCH, MED)	541.0000	0.0000000
DIST(BIOCH, MEDW)	671.0000	0.0000000
DIST(PHY, CC)	208.0000	0.0000000
DIST(PHY, BIOCH)	50.00000	0.0000000
DIST(PHY, PHY)	0.0000000	0.0000000
DIST(PHY, CS)	50.00000	0.0000000
DIST(PHY, STATS)	370.0000	0.0000000
DIST(PHY, MATHS)	319.0000	0.0000000
DIST(PHY, LAW)	642.0000	0.0000000
DIST(PHY, MBA)	597.0000	0.0000000
DIST(PHY, COMM)	997.0000	0.0000000
DIST(PHY, MED)	350.0000	0.0000000
DIST(PHY, MEDW)	456.0000	0.0000000

DIST(CS, CC)	206.0000	0.0000000
DIST(CS, BIOCH)	251.0000	0.0000000
DIST(CS, PHY)	50.00000	0.0000000
DIST(CS, CS)	0.0000000	0.0000000
DIST(CS, STATS)	320.0000	0.0000000
DIST(CS, MATHS)	269.0000	0.0000000
DIST(CS, LAW)	641.0000	0.0000000
DIST(CS, MBA)	596.0000	0.0000000
DIST(CS, COMM)	696.0000	0.0000000
DIST(CS, MED)	290.0000	0.0000000
DIST(CS, MEDW)	406.0000	0.0000000
DIST(STATS, CC)	442.0000	0.0000000
DIST(STATS, BIOCH)	571.0000	0.0000000
DIST(STATS, PHY)	370.0000	0.0000000
DIST(STATS, CS)	320.0000	0.0000000
DIST(STATS, STATS)	0.0000000	0.0000000
DIST(STATS, MATHS)	30.00000	0.0000000
DIST(STATS, LAW)	877.0000	0.0000000
DIST(STATS, MBA)	832.0000	0.0000000
DIST(STATS, COMM)	932.0000	0.0000000
DIST(STATS, MED)	610.0000	0.0000000
DIST(STATS, MEDW)	726.0000	0.0000000
DIST(MATHS, CC)	391.0000	0.0000000
DIST(MATHS, BIOCH)	521.0000	0.0000000
DIST(MATHS, PHY)	319.0000	0.0000000
DIST(MATHS, CS)	269.0000	0.0000000
DIST(MATHS, STATS)	30.00000	0.0000000
DIST(MATHS, MATHS)	0.0000000	0.0000000
DIST(MATHS, LAW)	826.0000	0.0000000
DIST(MATHS, MBA)	781.0000	0.0000000
DIST(MATHS, COMM)	881.0000	0.0000000
DIST(MATHS, MED)	559.0000	0.0000000
DIST(MATHS, MEDW)	675.0000	0.0000000
DIST(LAW, CC)	435.0000	0.0000000
DIST(LAW, BIOCH)	481.0000	0.0000000
DIST(LAW, PHY)	642.0000	0.0000000
DIST(LAW, CS)	641.0000	0.0000000
DIST(LAW, STATS)	961.0000	0.0000000
DIST(LAW, MATHS)	826.0000	0.0000000
DIST(LAW, LAW)	0.0000000	0.0000000
DIST(LAW, MBA)	110.0000	0.0000000
DIST(LAW, COMM)	30.00000	0.0000000
DIST(LAW, MED)	847.0000	0.0000000
DIST(LAW, MEDW)	963.0000	0.0000000
DIST(MBA, CC)	390.0000	0.0000000
DIST(MBA, BIOCH)	436.0000	0.0000000
DIST(MBA, PHY)	597.0000	0.0000000
DIST(MBA, CS)	596.0000	0.0000000
DIST(MBA, STATS)	916.0000	0.0000000
DIST(MBA, MATHS)	781.0000	0.0000000
DIST(MBA, LAW)	110.0000	0.0000000
DIST(MBA, MBA)	0.0000000	0.0000000
DIST(MBA, COMM)	80.00000	0.0000000
DIST(MBA, MED)	802.0000	0.0000000
DIST(MBA, MEDW)	918.0000	0.0000000
DIST(COMM, CC)	490.0000	0.0000000
DIST(COMM, BIOCH)	536.0000	0.0000000

DIST(COMM, PHY)	697.0000	0.0000000
DIST(COMM, CS)	696.0000	0.0000000
DIST(COMM, STATS)	932.0000	0.0000000
DIST(COMM, MATHS)	881.0000	0.0000000
DIST(COMM, LAW)	30.00000	0.0000000
DIST(COMM, MBA)	80.00000	0.0000000
DIST(COMM, COMM)	0.0000000	0.0000000
DIST(COMM, MED)	902.0000	0.0000000
DIST(COMM, MEDW)	1018.000	0.0000000
DIST(MED, CC)	412.0000	0.0000000
DIST(MED, BIOCH)	541.0000	0.0000000
DIST(MED, PHY)	340.0000	0.0000000
DIST(MED, CS)	290.0000	0.0000000
DIST(MED, STATS)	610.0000	0.0000000
DIST(MED, MATHS)	559.0000	0.0000000
DIST(MED, LAW)	847.0000	0.0000000
DIST(MED, MBA)	802.0000	0.0000000
DIST(MED, COMM)	902.0000	0.0000000
DIST(MED, MED)	0.0000000	0.0000000
DIST(MED, MEDW)	130.0000	0.0000000
DIST(MEDW, CC)	528.0000	0.0000000
DIST(MEDW, BIOCH)	671.0000	0.0000000
DIST(MEDW, PHY)	456.0000	0.0000000
DIST(MEDW, CS)	406.0000	0.0000000
DIST(MEDW, STATS)	726.0000	0.0000000
DIST(MEDW, MATHS)	675.0000	0.0000000
DIST(MEDW, LAW)	963.0000	0.0000000
DIST(MEDW, MBA)	918.0000	0.0000000
DIST(MEDW, COMM)	1018.000	0.0000000
DIST(MEDW, MED)	130.0000	0.0000000
DIST(MEDW, MEDW)	0.0000000	0.0000000
X(CC, CC)	0.0000000	0.0000000
X(CC, BIOCH)	1.000000	46.00000
X(CC, PHY)	0.0000000	207.0000
X(CC, CS)	0.0000000	206.0000
X(CC, STATS)	0.0000000	442.0000
X(CC, MATHS)	0.0000000	391.0000
X(CC, LAW)	0.0000000	435.0000
X(CC, MBA)	1.000000	390.0000
X(CC, COMM)	0.0000000	490.0000
X(CC, MED)	0.0000000	412.0000
X(CC, MEDW)	0.0000000	528.0000
X(BIOCH, CC)	0.0000000	46.00000
X(BIOCH, BIOCH)	0.0000000	0.0000000
X(BIOCH, PHY)	1.000000	50.00000
X(BIOCH, CS)	0.0000000	251.0000
X(BIOCH, STATS)	0.0000000	571.0000
X(BIOCH, MATHS)	0.0000000	521.0000
X(BIOCH, LAW)	0.0000000	481.0000
X(BIOCH, MBA)	0.0000000	435.0000
X(BIOCH, COMM)	0.0000000	536.0000
X(BIOCH, MED)	0.0000000	541.0000
X(BIOCH, MEDW)	0.0000000	671.0000
X(PHY, CC)	0.0000000	208.0000
X(PHY, BIOCH)	0.0000000	50.00000
X(PHY, PHY)	0.0000000	0.0000000
X(PHY, CS)	1.000000	50.00000

X(PHY, STATS)	0.0000000	370.0000
X(PHY, MATHS)	0.0000000	319.0000
X(PHY, LAW)	0.0000000	642.0000
X(PHY, MBA)	0.0000000	597.0000
X(PHY, COMM)	0.0000000	997.0000
X(PHY, MED)	0.0000000	350.0000
X(PHY, MEDW)	0.0000000	456.0000
X(CS, CC)	0.0000000	206.0000
X(CS, BIOCH)	0.0000000	251.0000
X(CS, PHY)	0.0000000	50.00000
X(CS, CS)	0.0000000	0.0000000
X(CS, STATS)	0.0000000	320.0000
X(CS, MATHS)	1.000000	269.0000
X(CS, LAW)	0.0000000	641.0000
X(CS, MBA)	0.0000000	596.0000
X(CS, COMM)	0.0000000	696.0000
X(CS, MED)	1.000000	290.0000
X(CS, MEDW)	0.0000000	406.0000
X(STATS, CC)	0.0000000	442.0000
X(STATS, BIOCH)	0.0000000	571.0000
X(STATS, PHY)	0.0000000	370.0000
X(STATS, CS)	0.0000000	320.0000
X(STATS, STATS)	0.0000000	0.0000000
X(STATS, MATHS)	0.0000000	30.00000
X(STATS, LAW)	0.0000000	877.0000
X(STATS, MBA)	0.0000000	832.0000
X(STATS, COMM)	0.0000000	932.0000
X(STATS, MED)	0.0000000	610.0000
X(STATS, MEDW)	0.0000000	726.0000
X(MATHS, CC)	0.0000000	391.0000
X(MATHS, BIOCH)	0.0000000	521.0000
X(MATHS, PHY)	0.0000000	319.0000
X(MATHS, CS)	0.0000000	269.0000
X(MATHS, STATS)	1.000000	30.00000
X(MATHS, MATHS)	0.0000000	0.0000000
X(MATHS, LAW)	0.0000000	826.0000
X(MATHS, MBA)	0.0000000	781.0000
X(MATHS, COMM)	0.0000000	881.0000
X(MATHS, MED)	0.0000000	559.0000
X(MATHS, MEDW)	0.0000000	675.0000
X(LAW, CC)	0.0000000	435.0000
X(LAW, BIOCH)	0.0000000	481.0000
X(LAW, PHY)	0.0000000	642.0000
X(LAW, CS)	0.0000000	641.0000
X(LAW, STATS)	0.0000000	961.0000
X(LAW, MATHS)	0.0000000	826.0000
X(LAW, LAW)	0.0000000	0.0000000
X(LAW, MBA)	0.0000000	110.0000
X(LAW, COMM)	0.0000000	30.00000
X(LAW, MED)	0.0000000	847.0000
X(LAW, MEDW)	0.0000000	963.0000
X(MBA, CC)	0.0000000	390.0000
X(MBA, BIOCH)	0.0000000	436.0000
X(MBA, PHY)	0.0000000	597.0000
X(MBA, CS)	0.0000000	596.0000
X(MBA, STATS)	0.0000000	916.0000
X(MBA, MATHS)	0.0000000	781.0000

X(MBA, LAW)	0.0000000	110.0000
X(MBA, MBA)	0.0000000	0.0000000
X(MBA, COMM)	1.000000	80.00000
X(MBA, MED)	0.0000000	802.0000
X(MBA, MEDW)	0.0000000	918.0000
X(COMM, CC)	0.0000000	490.0000
X(COMM, BIOCH)	0.0000000	536.0000
X(COMM, PHY)	0.0000000	697.0000
X(COMM, CS)	0.0000000	696.0000
X(COMM, STATS)	0.0000000	932.0000
X(COMM, MATHS)	0.0000000	881.0000
X(COMM, LAW)	1.000000	30.00000
X(COMM, MBA)	0.0000000	80.00000
X(COMM, COMM)	0.0000000	0.0000000
X(COMM, MED)	0.0000000	902.0000
X(COMM, MEDW)	0.0000000	1018.000
X(MED, CC)	0.0000000	412.0000
X(MED, BIOCH)	0.0000000	541.0000
X(MED, PHY)	0.0000000	340.0000
X(MED, CS)	0.0000000	290.0000
X(MED, STATS)	0.0000000	610.0000
X(MED, MATHS)	0.0000000	559.0000
X(MED, LAW)	0.0000000	847.0000
X(MED, MBA)	0.0000000	802.0000
X(MED, COMM)	0.0000000	902.0000
X(MED, MED)	0.0000000	0.0000000
X(MED, MEDW)	1.000000	130.0000
X(MEDW, CC)	0.0000000	528.0000
X(MEDW, BIOCH)	0.0000000	671.0000
X(MEDW, PHY)	0.0000000	456.0000
X(MEDW, CS)	0.0000000	406.0000
X(MEDW, STATS)	0.0000000	726.0000
X(MEDW, MATHS)	0.0000000	675.0000
X(MEDW, LAW)	0.0000000	963.0000
X(MEDW, MBA)	0.0000000	918.0000
X(MEDW, COMM)	0.0000000	1018.000
X(MEDW, MED)	0.0000000	130.0000
X(MEDW, MEDW)	0.0000000	0.0000000

Row	**Slack or Surplus**	**Dual Price**
1	0.0000000	0.0000000
2	1365.000	-1.000000
3	0.0000000	0.0000000
4	0.0000000	0.0000000
5	0.0000000	0.0000000
6	7.000000	0.0000000
7	5.000000	0.0000000
8	6.000000	0.0000000
9	7.000000	0.0000000
10	9.000000	0.0000000
11	8.000000	0.0000000
12	6.000000	0.0000000
13	5.000000	0.0000000
14	0.0000000	0.0000000
15	11.00000	0.0000000
16	0.0000000	0.0000000
17	0.0000000	0.0000000

18	6.000000	0.0000000
19	7.000000	0.0000000
20	8.000000	0.0000000
21	10.00000	0.0000000
22	9.000000	0.0000000
23	7.000000	0.0000000
24	6.000000	0.0000000
25	0.0000000	0.0000000
26	12.00000	0.0000000
27	11.00000	0.0000000
28	0.0000000	0.0000000
29	7.000000	0.0000000
30	0.0000000	0.0000000
31	9.000000	0.0000000
32	11.00000	0.0000000
33	10.00000	0.0000000
34	0.0000000	0.0000000
35	7.000000	0.0000000
36	0.0000000	0.0000000
37	14.00000	0.0000000
38	13.00000	0.0000000
39	12.00000	0.0000000
40	11.00000	0.0000000
41	0.0000000	0.0000000
42	11.00000	0.0000000
43	13.00000	0.0000000
44	12.00000	0.0000000
45	10.00000	0.0000000
46	9.000000	0.0000000
47	0.0000000	0.0000000
48	13.00000	0.0000000
49	12.00000	0.0000000
50	11.00000	0.0000000
51	0.0000000	0.0000000
52	0.0000000	0.0000000
53	10.00000	0.0000000
54	12.00000	0.0000000
55	11.00000	0.0000000
56	9.000000	0.0000000
57	8.000000	0.0000000
58	0.0000000	0.0000000
59	12.00000	0.0000000
60	11.00000	0.0000000
61	10.00000	0.0000000
62	9.000000	0.0000000
63	7.000000	0.0000000
64	8.000000	0.0000000
65	11.00000	0.0000000
66	0.0000000	0.0000000
67	8.000000	0.0000000
68	7.000000	0.0000000
69	0.0000000	0.0000000
70	0.0000000	0.0000000
71	9.000000	0.0000000
72	8.000000	0.0000000
73	7.000000	0.0000000
74	5.000000	0.0000000

75	6.000000	0.0000000
76	7.000000	0.0000000
77	0.0000000	0.0000000
78	6.000000	0.0000000
79	5.000000	0.0000000
80	0.0000000	0.0000000
81	11.00000	0.0000000
82	10.00000	0.0000000
83	9.000000	0.0000000
84	8.000000	0.0000000
85	6.000000	0.0000000
86	7.000000	0.0000000
87	0.0000000	0.0000000
88	0.0000000	0.0000000
89	7.000000	0.0000000
90	6.000000	0.0000000
91	0.0000000	0.0000000
92	13.00000	0.0000000
93	12.00000	0.0000000
94	11.00000	0.0000000
95	0.0000000	0.0000000
96	8.000000	0.0000000
97	9.000000	0.0000000
98	10.00000	0.0000000
99	12.00000	0.0000000
100	11.00000	0.0000000
101	0.0000000	0.0000000
102	0.0000000	0.0000000
103	14.00000	0.0000000
104	13.00000	0.0000000
105	12.00000	0.0000000
106	11.00000	0.0000000
107	9.000000	0.0000000
108	10.00000	0.0000000
109	11.00000	0.0000000
110	13.00000	0.0000000
111	12.00000	0.0000000
112	0.0000000	0.0000000
113	0.0000000	0.0000000
114	1.000000	0.0000000
115	0.0000000	0.0000000
116	8.000000	0.0000000
117	7.000000	0.0000000
118	5.000000	0.0000000
119	6.000000	0.0000000
120	7.000000	0.0000000
121	0.0000000	0.0000000
122	8.000000	0.0000000
123	6.000000	0.0000000
124	5.000000	0.0000000

Iterative Computation

(Minimal Spanning Tree) Network -II

Global optimal solution found at step: 300
Objective value: 489.5000
Branch count: 23

Variable	Value	Reduced Cost
N	8.000000	0.0000000
LVL(CED)	0.0000000	0.0000000
LVL(AS)	1.000000	0.0000000
LVL(ELCAL)	1.000000	0.0000000
LVL(ELNIC)	2.000000	0.0000000
LVL(COMP)	3.000000	0.0000000
LVL(UPOLY)	4.000000	0.0000000
LVL(BARCH)	1.000000	0.0000000
LVL(CHE_PAT)	2.000000	0.0000000
DIST(CED, CED)	0.0000000	0.0000000
DIST(CED, AS)	0.0000000	0.0000000
DIST(CED, ELCAL)	76.20000	0.0000000
DIST(CED, ELNIC)	131.1000	0.0000000
DIST(CED, COMP)	137.2000	0.0000000
DIST(CED, UPOLY)	297.2000	0.0000000
DIST(CED, BARCH)	137.2000	0.0000000
DIST(CED, CHE_PAT)	164.6000	0.0000000
DIST(AS, CED)	0.0000000	0.0000000
DIST(AS, AS)	0.0000000	0.0000000
DIST(AS, ELCAL)	76.20000	0.0000000
DIST(AS, ELNIC)	131.1000	0.0000000
DIST(AS, COMP)	137.2000	0.0000000
DIST(AS, UPOLY)	297.2000	0.0000000
DIST(AS, BARCH)	137.2000	0.0000000
DIST(AS, CHE_PAT)	164.6000	0.0000000
DIST(ELCAL, CED)	76.20000	0.0000000
DIST(ELCAL, AS)	76.20000	0.0000000
DIST(ELCAL, ELCAL)	0.0000000	0.0000000
DIST(ELCAL, ELNIC)	50.00000	0.0000000
DIST(ELCAL, COMP)	56.10000	0.0000000
DIST(ELCAL, UPOLY)	216.1000	0.0000000
DIST(ELCAL, BARCH)	213.4000	0.0000000
DIST(ELCAL, CHE_PAT)	240.8000	0.0000000
DIST(ELNIC, CED)	131.1000	0.0000000
DIST(ELNIC, AS)	131.1000	0.0000000
DIST(ELNIC, ELCAL)	50.00000	0.0000000
DIST(ELNIC, ELNIC)	0.0000000	0.0000000
DIST(ELNIC, COMP)	6.100000	0.0000000
DIST(ELNIC, UPOLY)	166.1000	0.0000000
DIST(ELNIC, BARCH)	268.3000	0.0000000
DIST(ELNIC, CHE_PAT)	295.7000	0.0000000
DIST(COMP, CED)	137.2000	0.0000000
DIST(COMP, AS)	137.2000	0.0000000
DIST(COMP, ELCAL)	56.10000	0.0000000
DIST(COMP, ELNIC)	6.100000	0.0000000
DIST(COMP, COMP)	0.0000000	0.0000000
DIST(COMP, UPOLY)	160.0000	0.0000000
DIST(COMP, BARCH)	274.4000	0.0000000
DIST(COMP, CHE_PAT)	301.8000	0.0000000

DIST(UPOLY, CED)	297.2000	0.0000000
DIST(UPOLY, AS)	297.2000	0.0000000
DIST(UPOLY, ELCAL)	216.1000	0.0000000
DIST(UPOLY, ELNIC)	166.1000	0.0000000
DIST(UPOLY, COMP)	160.0000	0.0000000
DIST(UPOLY, UPOLY)	0.0000000	0.0000000
DIST(UPOLY, BARCH)	434.4000	0.0000000
DIST(UPOLY, CHE_PAT)	461.8000	0.0000000
DIST(BARCH, CED)	137.2000	0.0000000
DIST(BARCH, AS)	137.2000	0.0000000
DIST(BARCH, ELCAL)	213.4000	0.0000000
DIST(BARCH, ELNIC)	268.3000	0.0000000
DIST(BARCH, COMP)	274.4000	0.0000000
DIST(BARCH, UPOLY)	434.4000	0.0000000
DIST(BARCH, BARCH)	0.0000000	0.0000000
DIST(BARCH, CHE_PAT)	60.00000	0.0000000
DIST(CHE_PAT, CED)	164.6000	0.0000000
DIST(CHE_PAT, AS)	164.6000	0.0000000
DIST(CHE_PAT, ELCAL)	240.8000	0.0000000
DIST(CHE_PAT, ELNIC)	295.7000	0.0000000
DIST(CHE_PAT, COMP)	301.8000	0.0000000
DIST(CHE_PAT, UPOLY)	461.8000	0.0000000
DIST(CHE_PAT, BARCH)	60.00000	0.0000000
DIST(CHE_PAT, CHE_PAT)	0.0000000	0.0000000
X(CED, CED)	0.0000000	0.0000000
X(CED, AS)	1.000000	0.0000000
X(CED, ELCAL)	1.000000	76.20000
X(CED, ELNIC)	0.0000000	131.1000
X(CED, COMP)	0.0000000	137.2000
X(CED, UPOLY)	0.0000000	297.2000
X(CED, BARCH)	1.000000	137.2000
X(CED, CHE_PAT)	0.0000000	164.6000
X(AS, CED)	0.0000000	0.0000000
X(AS, AS)	0.0000000	0.0000000
X(AS, ELCAL)	0.0000000	76.20000
X(AS, ELNIC)	0.0000000	131.1000
X(AS, COMP)	0.0000000	137.2000
X(AS, UPOLY)	0.0000000	297.2000
X(AS, BARCH)	0.0000000	137.2000
X(AS, CHE_PAT)	0.0000000	164.6000
X(ELCAL, CED)	0.0000000	76.20000
X(ELCAL, AS)	0.0000000	76.20000
X(ELCAL, ELCAL)	0.0000000	0.0000000
X(ELCAL, ELNIC)	1.000000	50.00000
X(ELCAL, COMP)	0.0000000	56.10000
X(ELCAL, UPOLY)	0.0000000	216.1000
X(ELCAL, BARCH)	0.0000000	213.4000
X(ELCAL, CHE_PAT)	0.0000000	240.8000
X(ELNIC, CED)	0.0000000	131.1000
X(ELNIC, AS)	0.0000000	131.1000
X(ELNIC, ELCAL)	0.0000000	50.00000
X(ELNIC, ELNIC)	0.0000000	0.0000000
X(ELNIC, COMP)	1.000000	6.100000
X(ELNIC, UPOLY)	0.0000000	166.1000
X(ELNIC, BARCH)	0.0000000	268.3000
X(ELNIC, CHE_PAT)	0.0000000	295.7000
X(COMP, CED)	0.0000000	137.2000

X(COMP, AS)	0.0000000	137.2000
X(COMP, ELCAL)	0.0000000	56.10000
X(COMP, ELNIC)	0.0000000	6.100000
X(COMP, COMP)	0.0000000	0.0000000
X(COMP, UPOLY)	1.000000	160.0000
X(COMP, BARCH)	0.0000000	274.4000
X(COMP, CHE_PAT)	0.0000000	301.8000
X(UPOLY, CED)	0.0000000	297.2000
X(UPOLY, AS)	0.0000000	297.2000
X(UPOLY, ELCAL)	0.0000000	216.1000
X(UPOLY, ELNIC)	0.0000000	166.1000
X(UPOLY, COMP)	0.0000000	160.0000
X(UPOLY, UPOLY)	0.0000000	0.0000000
X(UPOLY, BARCH)	0.0000000	434.4000
X(UPOLY, CHE_PAT)	0.0000000	461.8000
X(BARCH, CED)	0.0000000	137.2000
X(BARCH, AS)	0.0000000	137.2000
X(BARCH, ELCAL)	0.0000000	213.4000
X(BARCH, ELNIC)	0.0000000	268.3000
X(BARCH, COMP)	0.0000000	274.4000
X(BARCH, UPOLY)	0.0000000	434.4000
X(BARCH, BARCH)	0.0000000	0.0000000
X(BARCH, CHE_PAT)	1.000000	60.00000
X(CHE_PAT, CED)	0.0000000	164.6000
X(CHE_PAT, AS)	0.0000000	164.6000
X(CHE_PAT, ELCAL)	0.0000000	240.8000
X(CHE_PAT, ELNIC)	0.0000000	295.7000
X(CHE_PAT, COMP)	0.0000000	301.8000
X(CHE_PAT, UPOLY)	0.0000000	461.8000
X(CHE_PAT, BARCH)	0.0000000	60.00000
X(CHE_PAT, CHE_PAT)	0.0000000	0.0000000

Row	**Slack or Surplus**	**Dual Price**
1	0.0000000	0.0000000
2	489.5000	-1.000000
3	0.0000000	0.0000000
4	0.0000000	0.0000000
5	6.000000	0.0000000
6	5.000000	0.0000000
7	4.000000	0.0000000
8	3.000000	0.0000000
9	6.000000	0.0000000
10	5.000000	0.0000000
11	0.0000000	0.0000000
12	0.0000000	0.0000000
13	6.000000	0.0000000
14	0.0000000	0.0000000
15	4.000000	0.0000000
16	3.000000	0.0000000
17	6.000000	0.0000000
18	5.000000	0.0000000
19	0.0000000	0.0000000
20	8.000000	0.0000000
21	7.000000	0.0000000
22	0.0000000	0.0000000
23	0.0000000	0.0000000
24	4.000000	0.0000000

25	7.000000	0.0000000
26	6.000000	0.0000000
27	0.0000000	0.0000000
28	9.000000	0.0000000
29	8.000000	0.0000000
30	8.000000	0.0000000
31	0.0000000	0.0000000
32	0.0000000	0.0000000
33	8.000000	0.0000000
34	7.000000	0.0000000
35	0.0000000	0.0000000
36	10.00000	0.0000000
37	9.000000	0.0000000
38	9.000000	0.0000000
39	8.000000	0.0000000
40	0.0000000	0.0000000
41	9.000000	0.0000000
42	8.000000	0.0000000
43	0.0000000	0.0000000
44	0.0000000	0.0000000
45	6.000000	0.0000000
46	6.000000	0.0000000
47	5.000000	0.0000000
48	4.000000	0.0000000
49	3.000000	0.0000000
50	0.0000000	0.0000000
51	0.0000000	0.0000000
52	8.000000	0.0000000
53	7.000000	0.0000000
54	7.000000	0.0000000
55	6.000000	0.0000000
56	5.000000	0.0000000
57	4.000000	0.0000000
58	0.0000000	0.0000000
59	0.0000000	0.0000000
60	2.000000	0.0000000
61	0.0000000	0.0000000
62	0.0000000	0.0000000
63	5.000000	0.0000000
64	4.000000	0.0000000
65	3.000000	0.0000000
66	0.0000000	0.0000000
67	5.000000	0.0000000

Iterative Computation of the Shortest Path Problem for the COMPUTER CNETRE

PART - I

Global optimal solution found at step: 6256
Objective value: 2647.000
Branch count: 180

Variable	Value	Reduced Cost
N	11.00000	0.0000000
U(1)	0.0000000	0.0000000
U(2)	7.000000	0.0000000
U(3)	6.000000	0.0000000
U(4)	5.000000	0.0000000
U(5)	3.000000	0.0000000
U(6)	4.000000	0.0000000
U(7)	10.00000	0.0000000
U(8)	8.000000	0.0000000
U(9)	9.000000	0.0000000
U(10)	1.000000	0.0000000
U(11)	2.000000	0.0000000
DIST(1, 1)	0.0000000	0.0000000
DIST(1, 2)	46.00000	0.0000000
DIST(1, 3)	207.0000	0.0000000
DIST(1, 4)	206.0000	0.0000000
DIST(1, 5)	442.0000	0.0000000
DIST(1, 6)	391.0000	0.0000000
DIST(1, 7)	435.0000	0.0000000
DIST(1, 8)	390.0000	0.0000000
DIST(1, 9)	490.0000	0.0000000
DIST(1, 10)	412.0000	0.0000000
DIST(1, 11)	528.0000	0.0000000
DIST(2, 1)	46.00000	0.0000000
DIST(2, 2)	0.0000000	0.0000000
DIST(2, 3)	50.00000	0.0000000
DIST(2, 4)	251.0000	0.0000000
DIST(2, 5)	571.0000	0.0000000
DIST(2, 6)	521.0000	0.0000000
DIST(2, 7)	481.0000	0.0000000
DIST(2, 8)	435.0000	0.0000000
DIST(2, 9)	536.0000	0.0000000
DIST(2, 10)	541.0000	0.0000000
DIST(2, 11)	671.0000	0.0000000
DIST(3, 1)	208.0000	0.0000000
DIST(3, 2)	50.00000	0.0000000
DIST(3, 3)	0.0000000	0.0000000
DIST(3, 4)	50.00000	0.0000000
DIST(3, 5)	370.0000	0.0000000
DIST(3, 6)	319.0000	0.0000000
DIST(3, 7)	642.0000	0.0000000
DIST(3, 8)	597.0000	0.0000000
DIST(3, 9)	997.0000	0.0000000
DIST(3, 10)	350.0000	0.0000000
DIST(3, 11)	456.0000	0.0000000
DIST(4, 1)	206.0000	0.0000000
DIST(4, 2)	251.0000	0.0000000
DIST(4, 3)	50.00000	0.0000000
DIST(4, 4)	0.0000000	0.0000000
DIST(4, 5)	320.0000	0.0000000

DIST(4, 6)	269.0000	0.0000000
DIST(4, 7)	641.0000	0.0000000
DIST(4, 8)	596.0000	0.0000000
DIST(4, 9)	696.0000	0.0000000
DIST(4, 10)	290.0000	0.0000000
DIST(4, 11)	406.0000	0.0000000
DIST(5, 1)	442.0000	0.0000000
DIST(5, 2)	571.0000	0.0000000
DIST(5, 3)	370.0000	0.0000000
DIST(5, 4)	320.0000	0.0000000
DIST(5, 5)	0.0000000	0.0000000
DIST(5, 6)	30.00000	0.0000000
DIST(5, 7)	877.0000	0.0000000
DIST(5, 8)	832.0000	0.0000000
DIST(5, 9)	932.0000	0.0000000
DIST(5, 10)	610.0000	0.0000000
DIST(5, 11)	726.0000	0.0000000
DIST(6, 1)	391.0000	0.0000000
DIST(6, 2)	521.0000	0.0000000
DIST(6, 3)	319.0000	0.0000000
DIST(6, 4)	269.0000	0.0000000
DIST(6, 5)	30.00000	0.0000000
DIST(6, 6)	0.0000000	0.0000000
DIST(6, 7)	826.0000	0.0000000
DIST(6, 8)	781.0000	0.0000000
DIST(6, 9)	881.0000	0.0000000
DIST(6, 10)	559.0000	0.0000000
DIST(6, 11)	675.0000	0.0000000
DIST(7, 1)	435.0000	0.0000000
DIST(7, 2)	481.0000	0.0000000
DIST(7, 3)	642.0000	0.0000000
DIST(7, 4)	641.0000	0.0000000
DIST(7, 5)	961.0000	0.0000000
DIST(7, 6)	826.0000	0.0000000
DIST(7, 7)	0.0000000	0.0000000
DIST(7, 8)	110.0000	0.0000000
DIST(7, 9)	30.00000	0.0000000
DIST(7, 10)	847.0000	0.0000000
DIST(7, 11)	963.0000	0.0000000
DIST(8, 1)	390.0000	0.0000000
DIST(8, 2)	436.0000	0.0000000
DIST(8, 3)	597.0000	0.0000000
DIST(8, 4)	596.0000	0.0000000
DIST(8, 5)	916.0000	0.0000000
DIST(8, 6)	781.0000	0.0000000
DIST(8, 7)	110.0000	0.0000000
DIST(8, 8)	0.0000000	0.0000000
DIST(8, 9)	80.00000	0.0000000
DIST(8, 10)	802.0000	0.0000000
DIST(8, 11)	918.0000	0.0000000
DIST(9, 1)	490.0000	0.0000000
DIST(9, 2)	536.0000	0.0000000
DIST(9, 3)	697.0000	0.0000000
DIST(9, 4)	696.0000	0.0000000
DIST(9, 5)	932.0000	0.0000000
DIST(9, 6)	881.0000	0.0000000
DIST(9, 7)	30.00000	0.0000000

DIST(9, 8)	80.00000	0.0000000
DIST(9, 9)	0.0000000	0.0000000
DIST(9, 10)	902.0000	0.0000000
DIST(9, 11)	1018.000	0.0000000
DIST(10, 1)	412.0000	0.0000000
DIST(10, 2)	541.0000	0.0000000
DIST(10, 3)	340.0000	0.0000000
DIST(10, 4)	290.0000	0.0000000
DIST(10, 5)	610.0000	0.0000000
DIST(10, 6)	559.0000	0.0000000
DIST(10, 7)	847.0000	0.0000000
DIST(10, 8)	802.0000	0.0000000
DIST(10, 9)	902.0000	0.0000000
DIST(10, 10)	0.0000000	0.0000000
DIST(10, 11)	130.0000	0.0000000
DIST(11, 1)	528.0000	0.0000000
DIST(11, 2)	671.0000	0.0000000
DIST(11, 3)	456.0000	0.0000000
DIST(11, 4)	406.0000	0.0000000
DIST(11, 5)	726.0000	0.0000000
DIST(11, 6)	675.0000	0.0000000
DIST(11, 7)	963.0000	0.0000000
DIST(11, 8)	918.0000	0.0000000
DIST(11, 9)	1018.000	0.0000000
DIST(11, 10)	130.0000	0.0000000
DIST(11, 11)	0.0000000	0.0000000
X(1, 1)	0.0000000	0.0000000
X(1, 2)	0.0000000	46.00000
X(1, 3)	0.0000000	207.0000
X(1, 4)	0.0000000	206.0000
X(1, 5)	0.0000000	442.0000
X(1, 6)	0.0000000	391.0000
X(1, 7)	0.0000000	435.0000
X(1, 8)	0.0000000	390.0000
X(1, 9)	0.0000000	490.0000
X(1, 10)	1.000000	412.0000
X(1, 11)	0.0000000	528.0000
X(2, 1)	0.0000000	46.00000
X(2, 2)	0.0000000	0.0000000
X(2, 3)	0.0000000	50.00000
X(2, 4)	0.0000000	251.0000
X(2, 5)	0.0000000	571.0000
X(2, 6)	0.0000000	521.0000
X(2, 7)	0.0000000	481.0000
X(2, 8)	1.000000	435.0000
X(2, 9)	0.0000000	536.0000
X(2, 10)	0.0000000	541.0000
X(2, 11)	0.0000000	671.0000
X(3, 1)	0.0000000	208.0000
X(3, 2)	1.000000	50.00000
X(3, 3)	0.0000000	0.0000000
X(3, 4)	0.0000000	50.00000
X(3, 5)	0.0000000	370.0000
X(3, 6)	0.0000000	319.0000
X(3, 7)	0.0000000	642.0000
X(3, 8)	0.0000000	597.0000
X(3, 9)	0.0000000	997.0000

X(3, 10)	0.0000000	350.0000
X(3, 11)	0.0000000	456.0000
X(4, 1)	0.0000000	206.0000
X(4, 2)	0.0000000	251.0000
X(4, 3)	1.000000	50.00000
X(4, 4)	0.0000000	0.0000000
X(4, 5)	0.0000000	320.0000
X(4, 6)	0.0000000	269.0000
X(4, 7)	0.0000000	641.0000
X(4, 8)	0.0000000	596.0000
X(4, 9)	0.0000000	696.0000
X(4, 10)	0.0000000	290.0000
X(4, 11)	0.0000000	406.0000
X(5, 1)	0.0000000	442.0000
X(5, 2)	0.0000000	571.0000
X(5, 3)	0.0000000	370.0000
X(5, 4)	0.0000000	320.0000
X(5, 5)	0.0000000	0.0000000
X(5, 6)	1.000000	30.00000
X(5, 7)	0.0000000	877.0000
X(5, 8)	0.0000000	832.0000
X(5, 9)	0.0000000	932.0000
X(5, 10)	0.0000000	610.0000
X(5, 11)	0.0000000	726.0000
X(6, 1)	0.0000000	391.0000
X(6, 2)	0.0000000	521.0000
X(6, 3)	0.0000000	319.0000
X(6, 4)	1.000000	269.0000
X(6, 5)	0.0000000	30.00000
X(6, 6)	0.0000000	0.0000000
X(6, 7)	0.0000000	826.0000
X(6, 8)	0.0000000	781.0000
X(6, 9)	0.0000000	881.0000
X(6, 10)	0.0000000	559.0000
X(6, 11)	0.0000000	675.0000
X(7, 1)	1.000000	435.0000
X(7, 2)	0.0000000	481.0000
X(7, 3)	0.0000000	642.0000
X(7, 4)	0.0000000	641.0000
X(7, 5)	0.0000000	961.0000
X(7, 6)	0.0000000	826.0000
X(7, 7)	0.0000000	0.0000000
X(7, 8)	0.0000000	110.0000
X(7, 9)	0.0000000	30.00000
X(7, 10)	0.0000000	847.0000
X(7, 11)	0.0000000	963.0000
X(8, 1)	0.0000000	390.0000
X(8, 2)	0.0000000	436.0000
X(8, 3)	0.0000000	597.0000
X(8, 4)	0.0000000	596.0000
X(8, 5)	0.0000000	916.0000
X(8, 6)	0.0000000	781.0000
X(8, 7)	0.0000000	110.0000
X(8, 8)	0.0000000	0.0000000
X(8, 9)	1.000000	80.00000
X(8, 10)	0.0000000	802.0000
X(8, 11)	0.0000000	918.0000

X(9, 1)	0.0000000	490.0000
X(9, 2)	0.0000000	536.0000
X(9, 3)	0.0000000	697.0000
X(9, 4)	0.0000000	696.0000
X(9, 5)	0.0000000	932.0000
X(9, 6)	0.0000000	881.0000
X(9, 7)	1.000000	30.00000
X(9, 8)	0.0000000	80.00000
X(9, 9)	0.0000000	0.0000000
X(9, 10)	0.0000000	902.0000
X(9, 11)	0.0000000	1018.000
X(10, 1)	0.0000000	412.0000
X(10, 2)	0.0000000	541.0000
X(10, 3)	0.0000000	340.0000
X(10, 4)	0.0000000	290.0000
X(10, 5)	0.0000000	610.0000
X(10, 6)	0.0000000	559.0000
X(10, 7)	0.0000000	847.0000
X(10, 8)	0.0000000	802.0000
X(10, 9)	0.0000000	902.0000
X(10, 10)	0.0000000	0.0000000
X(10, 11)	1.000000	130.0000
X(11, 1)	0.0000000	528.0000
X(11, 2)	0.0000000	671.0000
X(11, 3)	0.0000000	456.0000
X(11, 4)	0.0000000	406.0000
X(11, 5)	1.000000	726.0000
X(11, 6)	0.0000000	675.0000
X(11, 7)	0.0000000	963.0000
X(11, 8)	0.0000000	918.0000
X(11, 9)	0.0000000	1018.000
X(11, 10)	0.0000000	130.0000
X(11, 11)	0.0000000	0.0000000

Row	Slack or Surplus	Dual Price
1	0.0000000	0.0000000
2	2647.000	-1.000000
3	0.0000000	0.0000000
4	0.0000000	0.0000000
5	16.00000	0.0000000
6	15.00000	0.0000000
7	14.00000	0.0000000
8	12.00000	0.0000000
9	13.00000	0.0000000
10	11.00000	0.0000000
11	17.00000	0.0000000
12	18.00000	0.0000000
13	0.0000000	0.0000000
14	11.00000	0.0000000
15	0.0000000	0.0000000
16	0.0000000	0.0000000
17	0.0000000	0.0000000
18	7.000000	0.0000000
19	5.000000	0.0000000
20	6.000000	0.0000000
21	12.00000	0.0000000
22	0.0000000	0.0000000

23	11.00000	0.0000000
24	3.000000	0.0000000
25	4.000000	0.0000000
26	0.0000000	0.0000000
27	0.0000000	0.0000000
28	0.0000000	0.0000000
29	0.0000000	0.0000000
30	6.000000	0.0000000
31	7.000000	0.0000000
32	13.00000	0.0000000
33	11.00000	0.0000000
34	12.00000	0.0000000
35	4.000000	0.0000000
36	5.000000	0.0000000
37	0.0000000	0.0000000
38	0.0000000	0.0000000
39	11.00000	0.0000000
40	0.0000000	0.0000000
41	7.000000	0.0000000
42	0.0000000	0.0000000
43	14.00000	0.0000000
44	12.00000	0.0000000
45	13.00000	0.0000000
46	5.000000	0.0000000
47	6.000000	0.0000000
48	0.0000000	0.0000000
49	0.0000000	0.0000000
50	13.00000	0.0000000
51	12.00000	0.0000000
52	11.00000	0.0000000
53	0.0000000	0.0000000
54	16.00000	0.0000000
55	14.00000	0.0000000
56	15.00000	0.0000000
57	7.000000	0.0000000
58	0.0000000	0.0000000
59	0.0000000	0.0000000
60	0.0000000	0.0000000
61	12.00000	0.0000000
62	11.00000	0.0000000
63	0.0000000	0.0000000
64	0.0000000	0.0000000
65	15.00000	0.0000000
66	13.00000	0.0000000
67	14.00000	0.0000000
68	6.000000	0.0000000
69	7.000000	0.0000000
70	0.0000000	0.0000000
71	0.0000000	0.0000000
72	6.000000	0.0000000
73	5.000000	0.0000000
74	4.000000	0.0000000
75	2.000000	0.0000000
76	3.000000	0.0000000
77	7.000000	0.0000000
78	0.0000000	0.0000000
79	0.0000000	0.0000000

80	1.000000	0.0000000
81	0.0000000	0.0000000
82	0.0000000	0.0000000
83	0.0000000	0.0000000
84	7.000000	0.0000000
85	6.000000	0.0000000
86	4.000000	0.0000000
87	5.000000	0.0000000
88	11.00000	0.0000000
89	0.0000000	0.0000000
90	2.000000	0.0000000
91	3.000000	0.0000000
92	0.0000000	0.0000000
93	0.0000000	0.0000000
94	7.000000	0.0000000
95	6.000000	0.0000000
96	5.000000	0.0000000
97	3.000000	0.0000000
98	4.000000	0.0000000
99	0.0000000	0.0000000
100	0.0000000	0.0000000
101	1.000000	0.0000000
102	2.000000	0.0000000
103	0.0000000	0.0000000
104	0.0000000	0.0000000
105	15.00000	0.0000000
106	14.00000	0.0000000
107	13.00000	0.0000000
108	11.00000	0.0000000
109	12.00000	0.0000000
110	18.00000	0.0000000
111	16.00000	0.0000000
112	17.00000	0.0000000
113	0.0000000	0.0000000
114	0.0000000	0.0000000
115	0.0000000	0.0000000
116	14.00000	0.0000000
117	13.00000	0.0000000
118	12.00000	0.0000000
119	0.0000000	0.0000000
120	11.00000	0.0000000
121	17.00000	0.0000000
122	15.00000	0.0000000
123	16.00000	0.0000000
124	0.0000000	0.0000000
125	3.000000	0.0000000
126	6.000000	0.0000000
127	4.000000	0.0000000
128	5.000000	0.0000000
129	5.000000	0.0000000
130	4.000000	0.0000000
131	7.000000	0.0000000
132	2.000000	0.0000000
133	6.000000	0.0000000
134	3.000000	0.0000000
135	0.0000000	0.0000000
136	0.0000000	0.0000000

137	2.000000	0.0000000
138	7.000000	0.0000000
139	1.000000	0.0000000
140	8.000000	0.0000000
141	0.0000000	0.0000000
142	0.0000000	0.0000000
143	8.000000	0.0000000
144	1.000000	0.0000000

Iterative Computation of the Shortest Path Problem for the COMPUTER CNETRE

PART - II

Global optimal solution found at step: 2594
Objective value: 951.3000
Branch count: 169

Variable	Value	Reduced Cost
N	8.000000	0.0000000
U(1)	0.0000000	0.0000000
U(2)	3.000000	0.0000000
U(3)	4.000000	0.0000000
U(4)	5.000000	0.0000000
U(5)	6.000000	0.0000000
U(6)	7.000000	0.0000000
U(7)	1.000000	0.0000000
U(8)	2.000000	0.0000000
DIST(1, 1)	0.0000000	0.0000000
DIST(1, 2)	0.0000000	0.0000000
DIST(1, 3)	76.20000	0.0000000
DIST(1, 4)	131.1000	0.0000000
DIST(1, 5)	137.2000	0.0000000
DIST(1, 6)	297.2000	0.0000000
DIST(1, 7)	137.2000	0.0000000
DIST(1, 8)	164.6000	0.0000000
DIST(2, 1)	0.0000000	0.0000000
DIST(2, 2)	0.0000000	0.0000000
DIST(2, 3)	76.20000	0.0000000
DIST(2, 4)	131.1000	0.0000000
DIST(2, 5)	137.2000	0.0000000
DIST(2, 6)	297.2000	0.0000000
DIST(2, 7)	137.2000	0.0000000
DIST(2, 8)	164.6000	0.0000000
DIST(3, 1)	76.20000	0.0000000
DIST(3, 2)	76.20000	0.0000000
DIST(3, 3)	0.0000000	0.0000000
DIST(3, 4)	50.00000	0.0000000
DIST(3, 5)	56.10000	0.0000000
DIST(3, 6)	216.1000	0.0000000
DIST(3, 7)	213.4000	0.0000000
DIST(3, 8)	240.8000	0.0000000

DIST(4, 1)	131.1000	0.0000000
DIST(4, 2)	131.1000	0.0000000
DIST(4, 3)	50.00000	0.0000000
DIST(4, 4)	0.0000000	0.0000000
DIST(4, 5)	6.100000	0.0000000
DIST(4, 6)	166.1000	0.0000000
DIST(4, 7)	268.3000	0.0000000
DIST(4, 8)	295.7000	0.0000000
DIST(5, 1)	137.2000	0.0000000
DIST(5, 2)	137.2000	0.0000000
DIST(5, 3)	56.10000	0.0000000
DIST(5, 4)	6.100000	0.0000000
DIST(5, 5)	0.0000000	0.0000000
DIST(5, 6)	160.0000	0.0000000
DIST(5, 7)	274.4000	0.0000000
DIST(5, 8)	301.8000	0.0000000
DIST(6, 1)	297.2000	0.0000000
DIST(6, 2)	297.2000	0.0000000
DIST(6, 3)	216.1000	0.0000000
DIST(6, 4)	166.1000	0.0000000
DIST(6, 5)	160.0000	0.0000000
DIST(6, 6)	0.0000000	0.0000000
DIST(6, 7)	434.4000	0.0000000
DIST(6, 8)	461.8000	0.0000000
DIST(7, 1)	137.2000	0.0000000
DIST(7, 2)	137.2000	0.0000000
DIST(7, 3)	213.4000	0.0000000
DIST(7, 4)	268.3000	0.0000000
DIST(7, 5)	274.4000	0.0000000
DIST(7, 6)	434.4000	0.0000000
DIST(7, 7)	0.0000000	0.0000000
DIST(7, 8)	60.00000	0.0000000
DIST(8, 1)	164.6000	0.0000000
DIST(8, 2)	164.6000	0.0000000
DIST(8, 3)	240.8000	0.0000000
DIST(8, 4)	295.7000	0.0000000
DIST(8, 5)	301.8000	0.0000000
DIST(8, 6)	461.8000	0.0000000
DIST(8, 7)	60.00000	0.0000000
DIST(8, 8)	0.0000000	0.0000000
X(1, 1)	0.0000000	0.0000000
X(1, 2)	0.0000000	0.0000000
X(1, 3)	0.0000000	76.20000
X(1, 4)	0.0000000	131.1000
X(1, 5)	0.0000000	137.2000
X(1, 6)	0.0000000	297.2000
X(1, 7)	1.000000	137.2000
X(1, 8)	0.0000000	164.6000
X(2, 1)	0.0000000	0.0000000
X(2, 2)	0.0000000	0.0000000
X(2, 3)	1.000000	76.20000
X(2, 4)	0.0000000	131.1000
X(2, 5)	0.0000000	137.2000
X(2, 6)	0.0000000	297.2000
X(2, 7)	0.0000000	137.2000
X(2, 8)	0.0000000	164.6000
X(3, 1)	0.0000000	76.20000

X(3, 2)	0.0000000	76.20000
X(3, 3)	0.0000000	0.0000000
X(3, 4)	1.000000	50.00000
X(3, 5)	0.0000000	56.10000
X(3, 6)	0.0000000	216.1000
X(3, 7)	0.0000000	213.4000
X(3, 8)	0.0000000	240.8000
X(4, 1)	0.0000000	131.1000
X(4, 2)	0.0000000	131.1000
X(4, 3)	0.0000000	50.00000
X(4, 4)	0.0000000	0.0000000
X(4, 5)	1.000000	6.100000
X(4, 6)	0.0000000	166.1000
X(4, 7)	0.0000000	268.3000
X(4, 8)	0.0000000	295.7000
X(5, 1)	0.0000000	137.2000
X(5, 2)	0.0000000	137.2000
X(5, 3)	0.0000000	56.10000
X(5, 4)	0.0000000	6.100000
X(5, 5)	0.0000000	0.0000000
X(5, 6)	1.000000	160.0000
X(5, 7)	0.0000000	274.4000
X(5, 8)	0.0000000	301.8000
X(6, 1)	1.000000	297.2000
X(6, 2)	0.0000000	297.2000
X(6, 3)	0.0000000	216.1000
X(6, 4)	0.0000000	166.1000
X(6, 5)	0.0000000	160.0000
X(6, 6)	0.0000000	0.0000000
X(6, 7)	0.0000000	434.4000
X(6, 8)	0.0000000	461.8000
X(7, 1)	0.0000000	137.2000
X(7, 2)	0.0000000	137.2000
X(7, 3)	0.0000000	213.4000
X(7, 4)	0.0000000	268.3000
X(7, 5)	0.0000000	274.4000
X(7, 6)	0.0000000	434.4000
X(7, 7)	0.0000000	0.0000000
X(7, 8)	1.000000	60.00000
X(8, 1)	0.0000000	164.6000
X(8, 2)	1.000000	164.6000
X(8, 3)	0.0000000	240.8000
X(8, 4)	0.0000000	295.7000
X(8, 5)	0.0000000	301.8000
X(8, 6)	0.0000000	461.8000
X(8, 7)	0.0000000	60.00000
X(8, 8)	0.0000000	0.0000000

Row	Slack or Surplus	Dual Price
1	0.0000000	0.0000000
2	951.3000	-1.000000
3	0.0000000	0.0000000
4	0.0000000	0.0000000
5	9.000000	0.0000000
6	10.00000	0.0000000
7	11.00000	0.0000000
8	12.00000	0.0000000

9	8.000000	0.0000000
10	0.0000000	0.0000000
11	8.000000	0.0000000
12	0.0000000	0.0000000
13	0.0000000	0.0000000
14	0.0000000	0.0000000
15	8.000000	0.0000000
16	9.000000	0.0000000
17	10.00000	0.0000000
18	4.000000	0.0000000
19	0.0000000	0.0000000
20	0.0000000	0.0000000
21	0.0000000	0.0000000
22	0.0000000	0.0000000
23	0.0000000	0.0000000
24	8.000000	0.0000000
25	9.000000	0.0000000
26	3.000000	0.0000000
27	4.000000	0.0000000
28	0.0000000	0.0000000
29	0.0000000	0.0000000
30	4.000000	0.0000000
31	0.0000000	0.0000000
32	0.0000000	0.0000000
33	8.000000	0.0000000
34	2.000000	0.0000000
35	3.000000	0.0000000
36	0.0000000	0.0000000
37	0.0000000	0.0000000
38	3.000000	0.0000000
39	4.000000	0.0000000
40	0.0000000	0.0000000
41	0.0000000	0.0000000
42	1.000000	0.0000000
43	2.000000	0.0000000
44	0.0000000	0.0000000
45	0.0000000	0.0000000
46	2.000000	0.0000000
47	3.000000	0.0000000
48	4.000000	0.0000000
49	0.0000000	0.0000000
50	0.0000000	0.0000000
51	1.000000	0.0000000
52	0.0000000	0.0000000
53	0.0000000	0.0000000
54	8.000000	0.0000000
55	9.000000	0.0000000
56	10.00000	0.0000000
57	11.00000	0.0000000
58	12.00000	0.0000000
59	0.0000000	0.0000000
60	0.0000000	0.0000000
61	0.0000000	0.0000000
62	0.0000000	0.0000000
63	8.000000	0.0000000
64	9.000000	0.0000000
65	10.00000	0.0000000

66	11.00000	0.0000000
67	0.0000000	0.0000000
68	4.000000	0.0000000
69	2.000000	0.0000000
70	3.000000	0.0000000
71	3.000000	0.0000000
72	2.000000	0.0000000
73	4.000000	0.0000000
74	1.000000	0.0000000
75	5.000000	0.0000000
76	0.0000000	0.0000000
77	0.0000000	0.0000000
78	0.0000000	0.0000000
79	0.0000000	0.0000000
80	5.000000	0.0000000
81	1.000000	0.0000000

ITERATIVE COMPUTATION OF TRAVELLING GAS SERVICE (AMU)

Global optimal solution found at step: 1239
Objective value: 4815.000
Branch count: 14

Variable	Value	Reduced Cost
N	13.00000	0.0000000
U(1)	0.0000000	0.0000000
U(2)	12.00000	0.0000000
U(3)	1.000000	0.0000000
U(4)	2.000000	0.0000000
U(5)	3.000000	0.0000000
U(6)	11.00000	0.0000000
U(7)	10.00000	0.0000000
U(8)	9.000000	0.0000000
U(9)	7.000000	0.0000000
U(10)	6.000000	0.0000000
U(11)	8.000000	0.0000000
U(12)	5.000000	0.0000000
U(13)	4.000000	0.0000000
DIST(1, 1)	0.0000000	0.0000000
DIST(1, 2)	550.0000	0.0000000
DIST(1, 3)	635.0000	0.0000000
DIST(1, 4)	775.0000	0.0000000
DIST(1, 5)	1050.000	0.0000000
DIST(1, 6)	665.0000	0.0000000
DIST(1, 7)	930.0000	0.0000000
DIST(1, 8)	1080.000	0.0000000
DIST(1, 9)	950.0000	0.0000000
DIST(1, 10)	1180.000	0.0000000
DIST(1, 11)	1170.000	0.0000000
DIST(1, 12)	1470.000	0.0000000
DIST(1, 13)	1560.000	0.0000000
DIST(2, 1)	550.0000	0.0000000
DIST(2, 2)	0.0000000	0.0000000

DIST(2, 3)	285.0000	0.0000000
DIST(2, 4)	425.0000	0.0000000
DIST(2, 5)	550.0000	0.0000000
DIST(2, 6)	315.0000	0.0000000
DIST(2, 7)	580.0000	0.0000000
DIST(2, 8)	730.0000	0.0000000
DIST(2, 9)	600.0000	0.0000000
DIST(2, 10)	830.0000	0.0000000
DIST(2, 11)	820.0000	0.0000000
DIST(2, 12)	1120.000	0.0000000
DIST(2, 13)	1210.000	0.0000000
DIST(3, 1)	635.0000	0.0000000
DIST(3, 2)	285.0000	0.0000000
DIST(3, 3)	0.0000000	0.0000000
DIST(3, 4)	140.0000	0.0000000
DIST(3, 5)	490.0000	0.0000000
DIST(3, 6)	400.0000	0.0000000
DIST(3, 7)	665.0000	0.0000000
DIST(3, 8)	815.0000	0.0000000
DIST(3, 9)	685.0000	0.0000000
DIST(3, 10)	915.0000	0.0000000
DIST(3, 11)	905.0000	0.0000000
DIST(3, 12)	1205.000	0.0000000
DIST(3, 13)	1050.000	0.0000000
DIST(4, 1)	775.0000	0.0000000
DIST(4, 2)	425.0000	0.0000000
DIST(4, 3)	140.0000	0.0000000
DIST(4, 4)	0.0000000	0.0000000
DIST(4, 5)	350.0000	0.0000000
DIST(4, 6)	540.0000	0.0000000
DIST(4, 7)	805.0000	0.0000000
DIST(4, 8)	955.0000	0.0000000
DIST(4, 9)	825.0000	0.0000000
DIST(4, 10)	1055.000	0.0000000
DIST(4, 11)	1045.000	0.0000000
DIST(4, 12)	1345.000	0.0000000
DIST(4, 13)	910.0000	0.0000000
DIST(5, 1)	1050.000	0.0000000
DIST(5, 2)	550.0000	0.0000000
DIST(5, 3)	490.0000	0.0000000
DIST(5, 4)	350.0000	0.0000000
DIST(5, 5)	0.0000000	0.0000000
DIST(5, 6)	665.0000	0.0000000
DIST(5, 7)	930.0000	0.0000000
DIST(5, 8)	1080.000	0.0000000
DIST(5, 9)	330.0000	0.0000000
DIST(5, 10)	560.0000	0.0000000
DIST(5, 11)	550.0000	0.0000000
DIST(5, 12)	850.0000	0.0000000
DIST(5, 13)	660.0000	0.0000000
DIST(6, 1)	665.0000	0.0000000
DIST(6, 2)	315.0000	0.0000000
DIST(6, 3)	400.0000	0.0000000
DIST(6, 4)	540.0000	0.0000000
DIST(6, 5)	665.0000	0.0000000
DIST(6, 6)	0.0000000	0.0000000
DIST(6, 7)	265.0000	0.0000000

DIST(6, 8)	415.0000	0.0000000
DIST(6, 9)	715.0000	0.0000000
DIST(6, 10)	945.0000	0.0000000
DIST(6, 11)	935.0000	0.0000000
DIST(6, 12)	1235.000	0.0000000
DIST(6, 13)	1125.000	0.0000000
DIST(7, 1)	930.0000	0.0000000
DIST(7, 2)	580.0000	0.0000000
DIST(7, 3)	665.0000	0.0000000
DIST(7, 4)	805.0000	0.0000000
DIST(7, 5)	930.0000	0.0000000
DIST(7, 6)	265.0000	0.0000000
DIST(7, 7)	0.0000000	0.0000000
DIST(7, 8)	150.0000	0.0000000
DIST(7, 9)	620.0000	0.0000000
DIST(7, 10)	850.0000	0.0000000
DIST(7, 11)	620.0000	0.0000000
DIST(7, 12)	1140.000	0.0000000
DIST(7, 13)	1280.000	0.0000000
DIST(8, 1)	1080.000	0.0000000
DIST(8, 2)	730.0000	0.0000000
DIST(8, 3)	815.0000	0.0000000
DIST(8, 4)	955.0000	0.0000000
DIST(8, 5)	1080.000	0.0000000
DIST(8, 6)	415.0000	0.0000000
DIST(8, 7)	150.0000	0.0000000
DIST(8, 8)	0.0000000	0.0000000
DIST(8, 9)	690.0000	0.0000000
DIST(8, 10)	690.0000	0.0000000
DIST(8, 11)	470.0000	0.0000000
DIST(8, 12)	1020.000	0.0000000
DIST(8, 13)	1160.000	0.0000000
DIST(9, 1)	950.0000	0.0000000
DIST(9, 2)	600.0000	0.0000000
DIST(9, 3)	685.0000	0.0000000
DIST(9, 4)	825.0000	0.0000000
DIST(9, 5)	330.0000	0.0000000
DIST(9, 6)	715.0000	0.0000000
DIST(9, 7)	620.0000	0.0000000
DIST(9, 8)	690.0000	0.0000000
DIST(9, 9)	0.0000000	0.0000000
DIST(9, 10)	230.0000	0.0000000
DIST(9, 11)	220.0000	0.0000000
DIST(9, 12)	520.0000	0.0000000
DIST(9, 13)	660.0000	0.0000000
DIST(10, 1)	1180.000	0.0000000
DIST(10, 2)	830.0000	0.0000000
DIST(10, 3)	915.0000	0.0000000
DIST(10, 4)	1055.000	0.0000000
DIST(10, 5)	560.0000	0.0000000
DIST(10, 6)	945.0000	0.0000000
DIST(10, 7)	850.0000	0.0000000
DIST(10, 8)	690.0000	0.0000000
DIST(10, 9)	230.0000	0.0000000
DIST(10, 10)	0.0000000	0.0000000
DIST(10, 11)	220.0000	0.0000000
DIST(10, 12)	290.0000	0.0000000

DIST(10, 13)	430.0000	0.0000000
DIST(11, 1)	1170.000	0.0000000
DIST(11, 2)	820.0000	0.0000000
DIST(11, 3)	905.0000	0.0000000
DIST(11, 4)	1045.000	0.0000000
DIST(11, 5)	550.0000	0.0000000
DIST(11, 6)	935.0000	0.0000000
DIST(11, 7)	620.0000	0.0000000
DIST(11, 8)	470.0000	0.0000000
DIST(11, 9)	220.0000	0.0000000
DIST(11, 10)	220.0000	0.0000000
DIST(11, 11)	0.0000000	0.0000000
DIST(11, 12)	550.0000	0.0000000
DIST(11, 13)	650.0000	0.0000000
DIST(12, 1)	1470.000	0.0000000
DIST(12, 2)	1120.000	0.0000000
DIST(12, 3)	1205.000	0.0000000
DIST(12, 4)	1345.000	0.0000000
DIST(12, 5)	850.0000	0.0000000
DIST(12, 6)	1235.000	0.0000000
DIST(12, 7)	1140.000	0.0000000
DIST(12, 8)	1020.000	0.0000000
DIST(12, 9)	520.0000	0.0000000
DIST(12, 10)	290.0000	0.0000000
DIST(12, 11)	550.0000	0.0000000
DIST(12, 12)	0.0000000	0.0000000
DIST(12, 13)	540.0000	0.0000000
DIST(13, 1)	1560.000	0.0000000
DIST(13, 2)	1210.000	0.0000000
DIST(13, 3)	1050.000	0.0000000
DIST(13, 4)	910.0000	0.0000000
DIST(13, 5)	660.0000	0.0000000
DIST(13, 6)	1125.000	0.0000000
DIST(13, 7)	1280.000	0.0000000
DIST(13, 8)	1160.000	0.0000000
DIST(13, 9)	660.0000	0.0000000
DIST(13, 10)	430.0000	0.0000000
DIST(13, 11)	650.0000	0.0000000
DIST(13, 12)	540.0000	0.0000000
DIST(13, 13)	0.0000000	0.0000000
X(1, 1)	0.0000000	0.0000000
X(1, 2)	0.0000000	550.0000
X(1, 3)	1.000000	635.0000
X(1, 4)	0.0000000	775.0000
X(1, 5)	0.0000000	1050.000
X(1, 6)	0.0000000	665.0000
X(1, 7)	0.0000000	930.0000
X(1, 8)	0.0000000	1080.000
X(1, 9)	0.0000000	950.0000
X(1, 10)	0.0000000	1180.000
X(1, 11)	0.0000000	1170.000
X(1, 12)	0.0000000	1470.000
X(1, 13)	0.0000000	1560.000
X(2, 1)	1.000000	550.0000
X(2, 2)	0.0000000	0.0000000
X(2, 3)	0.0000000	285.0000
X(2, 4)	0.0000000	425.0000

X(2, 5)	0.0000000	550.0000
X(2, 6)	0.0000000	315.0000
X(2, 7)	0.0000000	580.0000
X(2, 8)	0.0000000	730.0000
X(2, 9)	0.0000000	600.0000
X(2, 10)	0.0000000	830.0000
X(2, 11)	0.0000000	820.0000
X(2, 12)	0.0000000	1120.000
X(2, 13)	0.0000000	1210.000
X(3, 1)	0.0000000	635.0000
X(3, 2)	0.0000000	285.0000
X(3, 3)	0.0000000	0.0000000
X(3, 4)	1.000000	140.0000
X(3, 5)	0.0000000	490.0000
X(3, 6)	0.0000000	400.0000
X(3, 7)	0.0000000	665.0000
X(3, 8)	0.0000000	815.0000
X(3, 9)	0.0000000	685.0000
X(3, 10)	0.0000000	915.0000
X(3, 11)	0.0000000	905.0000
X(3, 12)	0.0000000	1205.000
X(3, 13)	0.0000000	1050.000
X(4, 1)	0.0000000	775.0000
X(4, 2)	0.0000000	425.0000
X(4, 3)	0.0000000	140.0000
X(4, 4)	0.0000000	0.0000000
X(4, 5)	1.000000	350.0000
X(4, 6)	0.0000000	540.0000
X(4, 7)	0.0000000	805.0000
X(4, 8)	0.0000000	955.0000
X(4, 9)	0.0000000	825.0000
X(4, 10)	0.0000000	1055.000
X(4, 11)	0.0000000	1045.000
X(4, 12)	0.0000000	1345.000
X(4, 13)	0.0000000	910.0000
X(5, 1)	0.0000000	1050.000
X(5, 2)	0.0000000	550.0000
X(5, 3)	0.0000000	490.0000
X(5, 4)	0.0000000	350.0000
X(5, 5)	0.0000000	0.0000000
X(5, 6)	0.0000000	665.0000
X(5, 7)	0.0000000	930.0000
X(5, 8)	0.0000000	1080.000
X(5, 9)	0.0000000	330.0000
X(5, 10)	0.0000000	560.0000
X(5, 11)	0.0000000	550.0000
X(5, 12)	0.0000000	850.0000
X(5, 13)	1.000000	660.0000
X(6, 1)	0.0000000	665.0000
X(6, 2)	1.000000	315.0000
X(6, 3)	0.0000000	400.0000
X(6, 4)	0.0000000	540.0000
X(6, 5)	0.0000000	665.0000
X(6, 6)	0.0000000	0.0000000
X(6, 7)	0.0000000	265.0000
X(6, 8)	0.0000000	415.0000
X(6, 9)	0.0000000	715.0000

X(6, 10)	0.0000000	945.0000
X(6, 11)	0.0000000	935.0000
X(6, 12)	0.0000000	1235.000
X(6, 13)	0.0000000	1125.000
X(7, 1)	0.0000000	930.0000
X(7, 2)	0.0000000	580.0000
X(7, 3)	0.0000000	665.0000
X(7, 4)	0.0000000	805.0000
X(7, 5)	0.0000000	930.0000
X(7, 6)	1.000000	265.0000
X(7, 7)	0.0000000	0.0000000
X(7, 8)	0.0000000	150.0000
X(7, 9)	0.0000000	620.0000
X(7, 10)	0.0000000	850.0000
X(7, 11)	0.0000000	620.0000
X(7, 12)	0.0000000	1140.000
X(7, 13)	0.0000000	1280.000
X(8, 1)	0.0000000	1080.000
X(8, 2)	0.0000000	730.0000
X(8, 3)	0.0000000	815.0000
X(8, 4)	0.0000000	955.0000
X(8, 5)	0.0000000	1080.000
X(8, 6)	0.0000000	415.0000
X(8, 7)	1.000000	150.0000
X(8, 8)	0.0000000	0.0000000
X(8, 9)	0.0000000	690.0000
X(8, 10)	0.0000000	690.0000
X(8, 11)	0.0000000	470.0000
X(8, 12)	0.0000000	1020.000
X(8, 13)	0.0000000	1160.000
X(9, 1)	0.0000000	950.0000
X(9, 2)	0.0000000	600.0000
X(9, 3)	0.0000000	685.0000
X(9, 4)	0.0000000	825.0000
X(9, 5)	0.0000000	330.0000
X(9, 6)	0.0000000	715.0000
X(9, 7)	0.0000000	620.0000
X(9, 8)	0.0000000	690.0000
X(9, 9)	0.0000000	0.0000000
X(9, 10)	0.0000000	230.0000
X(9, 11)	1.000000	220.0000
X(9, 12)	0.0000000	520.0000
X(9, 13)	0.0000000	660.0000
X(10, 1)	0.0000000	1180.000
X(10, 2)	0.0000000	830.0000
X(10, 3)	0.0000000	915.0000
X(10, 4)	0.0000000	1055.000
X(10, 5)	0.0000000	560.0000
X(10, 6)	0.0000000	945.0000
X(10, 7)	0.0000000	850.0000
X(10, 8)	0.0000000	690.0000
X(10, 9)	1.000000	230.0000
X(10, 10)	0.0000000	0.0000000
X(10, 11)	0.0000000	220.0000
X(10, 12)	0.0000000	290.0000
X(10, 13)	0.0000000	430.0000
X(11, 1)	0.0000000	1170.000

X(11, 2)	0.0000000	820.0000
X(11, 3)	0.0000000	905.0000
X(11, 4)	0.0000000	1045.000
X(11, 5)	0.0000000	550.0000
X(11, 6)	0.0000000	935.0000
X(11, 7)	0.0000000	620.0000
X(11, 8)	1.000000	470.0000
X(11, 9)	0.0000000	220.0000
X(11, 10)	0.0000000	220.0000
X(11, 11)	0.0000000	0.0000000
X(11, 12)	0.0000000	550.0000
X(11, 13)	0.0000000	650.0000
X(12, 1)	0.0000000	1470.000
X(12, 2)	0.0000000	1120.000
X(12, 3)	0.0000000	1205.000
X(12, 4)	0.0000000	1345.000
X(12, 5)	0.0000000	850.0000
X(12, 6)	0.0000000	1235.000
X(12, 7)	0.0000000	1140.000
X(12, 8)	0.0000000	1020.000
X(12, 9)	0.0000000	520.0000
X(12, 10)	1.000000	290.0000
X(12, 11)	0.0000000	550.0000
X(12, 12)	0.0000000	0.0000000
X(12, 13)	0.0000000	540.0000
X(13, 1)	0.0000000	1560.000
X(13, 2)	0.0000000	1210.000
X(13, 3)	0.0000000	1050.000
X(13, 4)	0.0000000	910.0000
X(13, 5)	0.0000000	660.0000
X(13, 6)	0.0000000	1125.000
X(13, 7)	0.0000000	1280.000
X(13, 8)	0.0000000	1160.000
X(13, 9)	0.0000000	660.0000
X(13, 10)	0.0000000	430.0000
X(13, 11)	0.0000000	650.0000
X(13, 12)	1.000000	540.0000
X(13, 13)	0.0000000	0.0000000

Row	Slack or Surplus	Dual Price
1	0.0000000	0.0000000
2	4815.000	-1.000000
3	0.0000000	0.0000000
4	0.0000000	0.0000000
5	13.00000	0.0000000
6	0.0000000	0.0000000
7	13.00000	0.0000000
8	14.00000	0.0000000
9	22.00000	0.0000000
10	21.00000	0.0000000
11	20.00000	0.0000000
12	18.00000	0.0000000
13	17.00000	0.0000000
14	19.00000	0.0000000
15	16.00000	0.0000000
16	15.00000	0.0000000
17	0.0000000	0.0000000

18	0.0000000	0.0000000
19	0.0000000	0.0000000
20	1.000000	0.0000000
21	2.000000	0.0000000
22	0.0000000	0.0000000
23	9.000000	0.0000000
24	8.000000	0.0000000
25	6.000000	0.0000000
26	5.000000	0.0000000
27	7.000000	0.0000000
28	4.000000	0.0000000
29	3.000000	0.0000000
30	0.0000000	0.0000000
31	0.0000000	0.0000000
32	22.00000	0.0000000
33	0.0000000	0.0000000
34	13.00000	0.0000000
35	21.00000	0.0000000
36	20.00000	0.0000000
37	19.00000	0.0000000
38	17.00000	0.0000000
39	16.00000	0.0000000
40	18.00000	0.0000000
41	15.00000	0.0000000
42	14.00000	0.0000000
43	0.0000000	0.0000000
44	0.0000000	0.0000000
45	21.00000	0.0000000
46	0.0000000	0.0000000
47	0.0000000	0.0000000
48	20.00000	0.0000000
49	19.00000	0.0000000
50	18.00000	0.0000000
51	16.00000	0.0000000
52	15.00000	0.0000000
53	17.00000	0.0000000
54	14.00000	0.0000000
55	13.00000	0.0000000
56	0.0000000	0.0000000
57	0.0000000	0.0000000
58	20.00000	0.0000000
59	9.000000	0.0000000
60	0.0000000	0.0000000
61	19.00000	0.0000000
62	18.00000	0.0000000
63	17.00000	0.0000000
64	15.00000	0.0000000
65	14.00000	0.0000000
66	16.00000	0.0000000
67	13.00000	0.0000000
68	0.0000000	0.0000000
69	0.0000000	0.0000000
70	0.0000000	0.0000000
71	0.0000000	0.0000000
72	1.000000	0.0000000
73	2.000000	0.0000000
74	3.000000	0.0000000

75	0.0000000	0.0000000
76	9.000000	0.0000000
77	7.000000	0.0000000
78	6.000000	0.0000000
79	8.000000	0.0000000
80	5.000000	0.0000000
81	4.000000	0.0000000
82	0.0000000	0.0000000
83	0.0000000	0.0000000
84	13.00000	0.0000000
85	2.000000	0.0000000
86	3.000000	0.0000000
87	4.000000	0.0000000
88	0.0000000	0.0000000
89	0.0000000	0.0000000
90	8.000000	0.0000000
91	7.000000	0.0000000
92	9.000000	0.0000000
93	6.000000	0.0000000
94	5.000000	0.0000000
95	0.0000000	0.0000000
96	0.0000000	0.0000000
97	14.00000	0.0000000
98	3.000000	0.0000000
99	4.000000	0.0000000
100	5.000000	0.0000000
101	13.00000	0.0000000
102	0.0000000	0.0000000
103	9.000000	0.0000000
104	8.000000	0.0000000
105	0.0000000	0.0000000
106	7.000000	0.0000000
107	6.000000	0.0000000
108	0.0000000	0.0000000
109	0.0000000	0.0000000
110	16.00000	0.0000000
111	5.000000	0.0000000
112	6.000000	0.0000000
113	7.000000	0.0000000
114	15.00000	0.0000000
115	14.00000	0.0000000
116	13.00000	0.0000000
117	0.0000000	0.0000000
118	0.0000000	0.0000000
119	9.000000	0.0000000
120	8.000000	0.0000000
121	0.0000000	0.0000000
122	0.0000000	0.0000000
123	17.00000	0.0000000
124	6.000000	0.0000000
125	7.000000	0.0000000
126	8.000000	0.0000000
127	16.00000	0.0000000
128	15.00000	0.0000000
129	14.00000	0.0000000
130	0.0000000	0.0000000
131	13.00000	0.0000000

132	0.0000000	0.0000000
133	9.000000	0.0000000
134	0.0000000	0.0000000
135	0.0000000	0.0000000
136	15.00000	0.0000000
137	4.000000	0.0000000
138	5.000000	0.0000000
139	6.000000	0.0000000
140	14.00000	0.0000000
141	13.00000	0.0000000
142	0.0000000	0.0000000
143	0.0000000	0.0000000
144	9.000000	0.0000000
145	8.000000	0.0000000
146	7.000000	0.0000000
147	0.0000000	0.0000000
148	0.0000000	0.0000000
149	18.00000	0.0000000
150	7.000000	0.0000000
151	8.000000	0.0000000
152	9.000000	0.0000000
153	17.00000	0.0000000
154	16.00000	0.0000000
155	15.00000	0.0000000
156	13.00000	0.0000000
157	0.0000000	0.0000000
158	14.00000	0.0000000
159	0.0000000	0.0000000
160	0.0000000	0.0000000
161	0.0000000	0.0000000
162	19.00000	0.0000000
163	8.000000	0.0000000
164	9.000000	0.0000000
165	0.0000000	0.0000000
166	18.00000	0.0000000
167	17.00000	0.0000000
168	16.00000	0.0000000
169	14.00000	0.0000000
170	13.00000	0.0000000
171	15.00000	0.0000000
172	0.0000000	0.0000000
173	0.0000000	0.0000000
174	0.0000000	0.0000000
175	0.0000000	0.0000000
176	0.0000000	0.0000000
177	10.00000	0.0000000
178	1.000000	0.0000000
179	9.000000	0.0000000
180	2.000000	0.0000000
181	1.000000	0.0000000
182	10.00000	0.0000000
183	2.000000	0.0000000
184	9.000000	0.0000000
185	3.000000	0.0000000
186	8.000000	0.0000000
187	5.000000	0.0000000
188	6.000000	0.0000000

189	6.000000	0.0000000
190	5.000000	0.0000000
191	4.000000	0.0000000
192	7.000000	0.0000000
193	7.000000	0.0000000
194	4.000000	0.0000000
195	8.000000	0.0000000
196	3.000000	0.0000000

About LINGO Software

APPENDIX - C

LINGO, The Software Used:

LINGO is a computer software which can solve both linear as well as non-linear programming problems. The process of solving a mathematical program requires a large number of calculations and is, therefore, best performed by a computer programme. The main purpose of LINGO is to allow its user to input a model formulation, and in return to get the solution. The software assesses the correctness or appropriateness of the formulation based on the solution, and then quickly makes minor modifications to the formulation and repeats the process. LINGO features a wide range of commands, any of which may be invoked at any time. LINGO checks whether a particular command makes sense in a particular context. The most powerful feature of LINGO is its ability to model a large system. LINGO has a number of links that allow the user to import data from spreadsheet and export solutions backout to spreadsheets. These links include simple file link to Excel and Lotus, real-time OLE links to Excel.

There are two versions of LINGO, (a) A Windows – Specific version and (b) a Generic, text-based version. The text-based version runs under most popular operating system, including Unix and Linux. The problems taken in this dissertation are solved by Window based LINGO.

When developing a model in LINGO, it helps to understand how the model is processed internally by the LINGO solver.

LINGO has four solvers, it uses to solve different types of models. These solvers are:

¬ a direct solver
¬ a linear solver
¬ a non-liner solver, and
¬ a branch-and-bound Manager.

The LINGO solvers, unlike solvers sold with other modeling language, are all part of the same programe. In other words, they are linked directly to the modeling language. This allows LINGO to pass data to its solvers directly through memory, rather than through intermediate files. Direct link to LINGO's solver also minimizes capability problems between the modeling language component and the solver components.

References

REFERENCES

1. Balakrishnan, A.V. & Lucien, W.N. (1964): *Computing Methods in Optimization Problems,* Academic Press, New York.

2. Banks, J., John, S.C.II, and L.N. Barry (1999): *Discrete Event System Simulation,* Prentice Hall of India, New Delhi.

3. Bazara, M. S. and Jarvin, J. J. (1977): *Linear Programming and Network Flows,* John Wiley & Sons, New York.

4. Dolan, A. and Aldous, J. (1993): *Networks and Algorithms: An Introduction,* John Wiley & Sons, Chichester.

5. Eric, D. (1962): *A Guide to Operations Research,* Mithun & Co. Ltd. London.

6. Flouds, L.R. (1981): *Optimization Techniques: An Introduction.* Springer – Verlag, New York.

7. Fredrick, S.H. and Lieberman, G.J. (1991): *Introduction to Mathematical Programming,* McGraw – Hill Pub. Company, New Delhi, India.

8. Gillette, B.E. (1976): *Introduction to operations Research: A Computer-Oriented Algorithmic approach,* Tata McGraw Hill, New Delhi.

9. Gupta, S.K. (1985): *Linear Programming and Network Models,* EWP, New Delhi.

10. Hadly, G. (1969): *Linear Programming,* Oxford & IBH Publishing Co. Pvt. Ltd., New Delhi.

11. Optimization Modeling with LINGO, (2002), Lindo System Int., USA.

12. Kombo, N.S. (1984): *Mathematical Programming Techniques,* EW Press, Pvt. Ltd., New Delhi, India.

13. Paul, B., Bennett, L. F., and Linus, E.S. (1983): *A Guide to Simulation,* Springer – Verlag, New York, USA.

14. Peter, A.S., (1978): *Optimization of Transport Networks,* John Wiley & Sons, London.

15. Press, W.H., Teakolsky, S.A., Vetterling, W.T. and Flannery, B.R. (2000): *Numerical Recipes in FORTRAN: The Art of Scientific Computing,* Cambridge University Press.

16. Ripley, B.D. (1987): *Stochastic Simulation,* John Wiley & Sons, New York.

17. Rockfeller, R.T. (1984): *Network Flows and Monotropic Optimization*, John Wiley & Sons, New York.

18. Rubinstein, R.Y. (1981): *Simulation and Monte-Carlo Method*, John Wiley & Sons, New York.

19. Russel, L. A., and Sasieni, M.W. (1984): *Fundamentals of Operations Research*, Wiley Eastern limited, New Delhi.

20. Sarieni, M., Yaspan, A., and Friedman, L. (1959): *Operations Research: Methods and Problems*, John Wiley & Sons, London.

21. Shaghil, m. and Mushtque, M. (1993): *Network Management Technique*, Printwell, Jaipur (India).

22. Smith, K. s. (1982): *Network Optimizatioin Practice; A Computational Guide*, Ellis Harwood limited, Chichester, U.K.

23. Swarup, K., Gupta, P.K. and Man Mohan (1995): *Operations Research*, S. Chand & Sons, New Delhi.

24. Taha, H. A. (1999): *Operations Research: An Introduction*, Prentice-Hall of India, Pvt. Lt., New Delhi.

25. Venkatakrishnan, S.D. (1992): *Operations Research: Principles and Problems*, Keerti Publishing House (P) Ltd. Coimbatore.